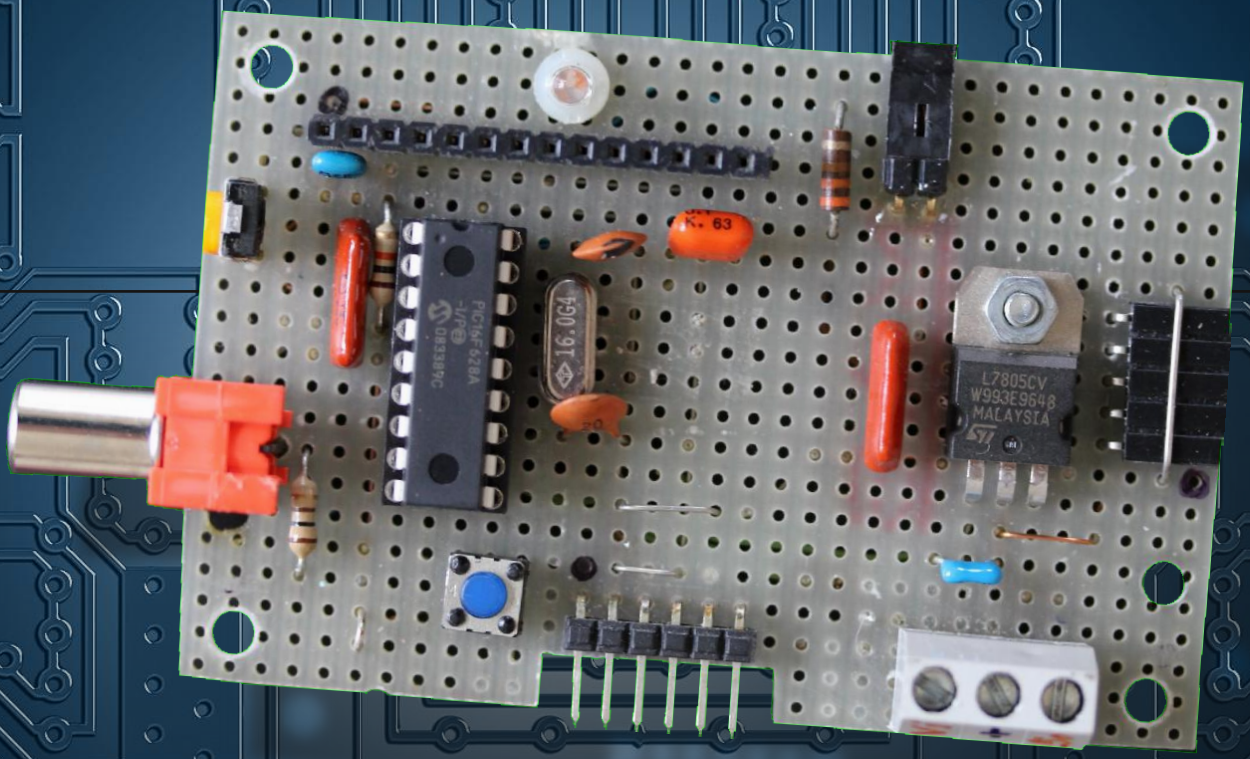
leggi Fare Elettronica su
cloud

fare elettronica
n. 365/366 - Aprile/Maggio 2016

Ricezione del segnale DCF77 con il PIC16F887



GENERATORE DI FREQUENZA CON UN PIC

- Aggiungiamo una Camera al Banana Pi
- Misuratore di resistenza interna batterie (2)
- Orologio-sveglia con Arduino
- Arduino "OPTO", l'iperbole della autocostruzione

La più ampia selezione dei prodotti più innovativi.
Più di 4 milioni di prodotti di oltre 500 produttori.
mouser.com
Distributore autorizzato
Visita mouser.com

Fare Elettronica n. 365/366 - Aprile/Maggio 2016
SOMMARIO

3 **L'Editoriale**
di Giovanni Di Maria

4 **AGGIUNGIAMO UNA CAMERA AL BANANA PI**
di Ivan Scordato
Come collegare e configurare un ottimo modulo camera al nostro amato Banana Pi, per realizzare un'infinità di progetti interessanti.

10 **MISURATORE DI RESISTENZA INTERNA BATTERIE (2)**
di Grazia Ancona
Seconda parte dell'articolo dedicato a coloro che hanno l'esigenza di verificare in dettaglio lo stato di salute di singoli elementi o di interi pack.

17 **RICEZIONE E DECODIFICA DEL DCF77 UTILIZZANDO IL TIMER1 DEL PIC16F887 DELLA MICROCHIP**
di Alberto Trasimeni
Tutte le tecniche e gli algoritmi per ricevere il famoso e potente segnale orario tedesco con un microcontrollore PIC.

30 **GENERATORE DI FREQUENZA CON UN PIC**
di Giovanni Carrera
Argomento di questo progetto è un generatore di onda quadra la cui frequenza, impostata da un encoder, varia da 4 Hz a 20 kHz.

41 **OROLOGIO-SVEGLIA CON ARDUINO**
di Girolamo D'Orio
Ecco un orologio-sveglia che non teme i blackout momentanei. Mantiene in memoria l'orario corrente, di allarme e lo stato di attivazione sveglia.

50 **ARDUINO "OPTO", L'IPERBOLE DELLA AUTOCOSTRUZIONE**
di Roberto Vallini
In questo articolo tratteremo approfonditamente il tema dell'autocostruzione, rapportata alle nuove tecnologie embended. La stazione saldante andrà in disuso? Addio "Millefori"? Forse sì, ma per il momento, c'è chi, come un reduce di guerra, porta alta la bandiera del suo reggimento e difende "l'antico profumo" di stagno fuso.

Novità! da questo numero gli articoli vengono categorizzati seguendo la legenda:

P Progetti

S Smart Home

D Divulgazione scientifica

L'Editoriale

di *Giovanni Di Maria*

Grandi novità in casa Fare Elettronica

Fare Elettronica è partita proprio alla grande e il 2016 inizia con una grande notizia. Focalizzando soprattutto l'attenzione alla comunicazione, IEcloud, la community dei progettisti elettronici e di Fare Elettronica, è stata acquisita da ***Tecnoimprese / Assodel****. Si tratta di un passo molto importante in quanto i due canali si aggiungono al sito Fortronic.net, alla rivista A&V Elettronica, agli eventi EW&D Fortronic, Power Fortronic e Illuminotronica, completando così la copertura del target di tecnici, progettisti, decision maker e buyer dell'elettronica. Per maggiori informazioni si può consultare il seguente comunicato stampa all'indirizzo:*
http://assodel.it/piu-comunicazione-nellelettronica/
Mi sento di augurare una proficua attività a tutti gli Autori. La rivista di Aprile/Maggio 2016 presenta, come sempre, tanti articoli interessanti e molte novità saranno proposte nei prossimi numeri. Grazia Ancona conclude l'articolo sul misuratore di resistenza interna per batterie. Ivan Scordato illustra come collegare un modulo Camera al Banana Pi, per acquisire video di buona qualità. Alberto Trasimeni spiega come ricevere il segnale orario tedesco con un PIC. Giovanni Carrera presenta un interessante progetto per generare frequenze con un microcontrollore. Girolamo D'Orio propone una prima versione di orologio con Arduino, promettendo un ampio approfondimento in uno dei prossimi numeri. Infine Roberto Vallini fa il punto della situazione dell'elettronica di ieri e di oggi con una serie si interessanti esperimenti. Gli articoli, a partire da questo numero, saranno contrassegnati con dei bollini colorati, il cui significato può essere desunto nell'indice. Buona lettura a tutti.

Giovanni Di Maria

AGGIUNGIAMO UN MODULO CAMERA AL NOSTRO BANANA PI

di Ivan Scordato

Negli articoli precedenti abbiamo visto dettagliatamente in cosa consiste il Banana Pi, come funziona e come configurarlo.
Oggi invece, come già accennato, trattiamo un argomento che può suscitare l'interesse di molti: vedremo come collegare e configurare un ottimo modulo camera al nostro amato Banana Pi, potendo realizzare così un'infinità di progetti interessanti.

Molto più spesso di quanto si possa pensare, nel momento in cui si sta progettando qualcosa che necessità di acquisire video di buona qualità, come ad esempio un robot rover o un drone, si commette un errore. Per cercare di risparmiare, o chissà per quale motivo, si decide di utilizzare una comunissima webcam come fonte di acquisizione, senza considerare che come soluzione non risulta essere delle migliori. Questo perchè, quando si vuole creare qualcosa di funzionale con quel pizzico di valore aggiunto che ogni vero Maker ama dare alle proprie creazioni, bisogna utilizzare dei prodotti adatti al proprio scopo che siano di buona qualità. Cercando online un modulo camera per il Banana Pi che rispecchi questi parametri, e che di conseguenza possegga delle buone caratteristiche, ho trovato quello che ritengo essere un accessorio molto interessante dal costo contenuto, cioè il modulo Camera con sensore di

Tabella 1: Caratteristiche	
Sensore Immagine	Omnivision 5640 CMOS - sensore di immagine in un modulo auto-focus con filtro IR integrato
Risoluzione	5 Megapixel
Formato matrice	2592x1944
Max frame rate	1080P 30fps 24Mhz
Formati immagine	JPEG PNG YUV420 RGB888
Formati video	Raw h.264
Connettore per Banana Pi	Interfaccia CSI-0 – Connettore40 pin FPC
Funzioni di controllo immagine	Controllo dell'esposizione automatico (AEC) Bilanciamento del bianco automatico (AWB) Calibrazione livello dei neri automatica (ABLC) Banda dei filtri automatica Funzione specchio e capovolto
Temperatura di funzionamento	Funzionamento: -30°C a 70°C Immagini stabili: 0°C a 50°C
Dimensioni lenti	1/4”
Dimensione	36x32x10 mm
Indirizzo I2C	0x78
Peso	8g

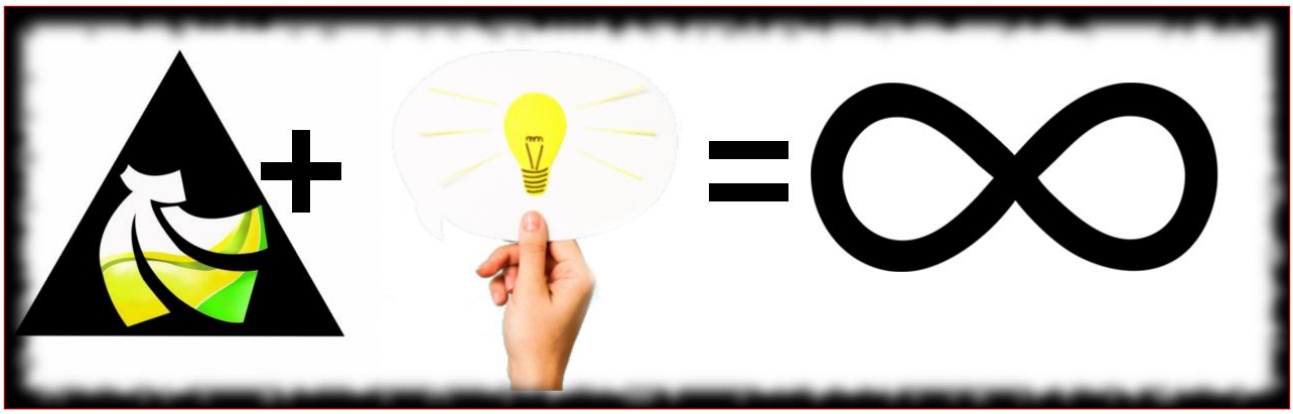


immagine OV5640 da 5 Megapixel. La camera in questione si presenta come quella di figura 2. E le caratteristiche del modulo sono riportate in tabella 1. Caratteristiche niente male per un modulo camera dal peso di 8g e che può essere facilmente agganciato a quello che vogliamo. Adesso vedremo cosa bisogna fare per collegarlo e settarlo sul Banana Pi.

Collegamento Hardware

Per effettuare questo collegamento è necessario utilizzare il cavetto flat dato in dotazione.
Connettere questo modulo camera al nostro Banana Pi è davvero semplice, infatti basterà collegarlo al connettore CON1 (Camera Serial Interface) presente sulla scheda, un connettore FPC 40 pin che può essere utilizzato per collegare un modulo camera con mappature appropriate. Prima di tutto bisogna girare il modulo con l'obiettivo puntato verso l'alto e quindi anche con il connettore in alto. Vedrete che è numerato da 0 a 40 da destra verso sinistra. Adesso, sempre con molta delicatezza, bisogna sollevare la linguetta nera del connettore senza forzarla, facendole assumere al massimo un angolo di 90°; inserire il cavetto flat con I contatti rivolti verso il basso e abbassare di nuovo la linguetta nera, in modo da tenere fermo il cavetto. Maneggiando questo modulo, quello che non posso finire di ripetermi è che le dimensioni ridottee del Banana Pi e del modulo Camera, pongono le basi essenziali per realizzare dei progetti che necessitano di componenti con una dimensione ridotta, come ad esempio un sistema di navigazione per robot e per la rilevazione di immagini ad alta quota con dei palloni sonda.

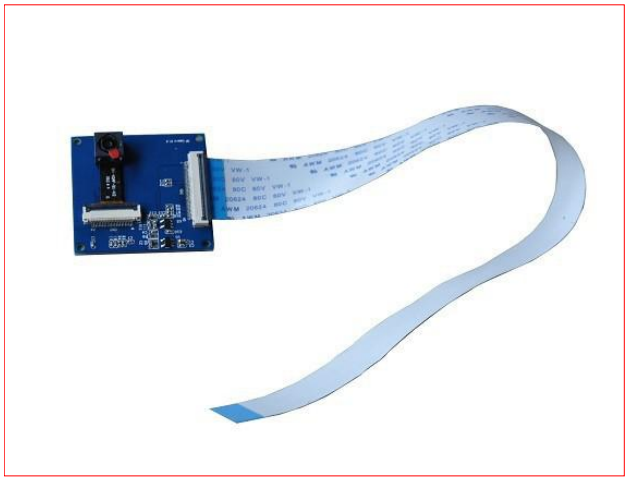


Figura 2: Modulo camera

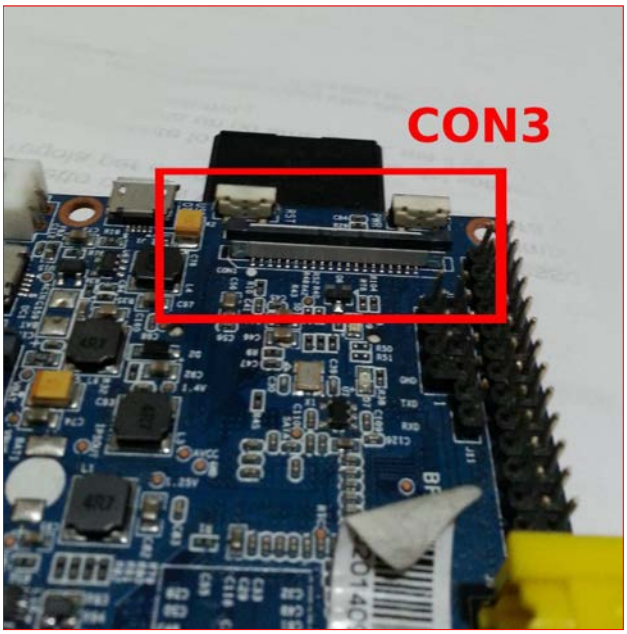


Figura 3: CON1 sul Banana Pi

ILLUMINOTRONICA

HUMAN CENTRIC LIGHTING TECHNOLOGIES

PADOVA 6/8 OTTOBRE 2016

i.p.

L'evento per chi fa, produce, genera, "crea" luce, per chi la utilizza, la modifica, la installa.

Il know-how del LED, dalla sua emissione, alimentazione, amplificazione, gestione, per chi lo vuole applicare e plasmare per le sue e altrui esigenze.

Tutte le tecnologie e le soluzioni integrate di luce, domotica e sicurezza per la smart home.

ILLUMINOTRONICA

Alla sua quinta edizione in Italia, è l'unico evento professionale a proporre il coordinamento di macroaree, prodotti e applicazioni, contenuti e tipologie d'interesse convergenti su:

LED & LIGHTING

Tecnologie & design nel dialogo-incontro
tra chi fornisce tecnologie e chi produce corpi e soluzioni

SMART HUT

Sistemi e soluzioni hw/sw per ottimizzare
applicazioni, sicurezza, efficienza nella domotica del "visibile"

ELETTRO & MICRO

Elettronica (componenti e sistemi)
per alimentare, assemblare, pilotare prodotti e soluzioni

PROMOSSO DA:

Associazione Italiana
Distretti Elettronica

assodel

IN PARTNERSHIP CON:

Consorzio Elettrimpex
Lumen International

LUMEN
CONSORZIO ELINT

Codega 2016

IV° EDIZIONE
INTERNATIONAL LIGHTING DESIGN PRIZE
VENEZIA
07 OTTOBRE

CONTATTI:

Tel. +39 02 210.111.236
segreteria@illuminotronica.it

TECNO IMPRESE

VUOI PARTECIPARE?
RICHIEDI LA DOMANDA DI ADESIONE
http://illuminotronica.it/exhibitors

www.illuminotronica.it

Figura 4: Modulo collegato

Configurazione Software

Per i vari test che ho eseguito, ho utilizzato il sistema operativo Lubuntu, ma naturalmente ognuno è libero di utilizzare quello che preferisce. Dopo avere collegato fisicamente la camera al Banana Pi, il primo passo da fare per utilizzarla è quello di installare i driver necessari per il suo interfacciamento, digitando i seguenti comandi direttamente sulla riga di comando o tramite il terminale:

```
sudo modprobe ov5640

sudo modprobe sun4i_csi
```

In questo modo, dopo aver digitato la password per eseguire l'operazione come superuser, basterà confermare lo scaricamento dei driver. Se per caso il secondo comando (modprobe sun4i_csi) non dovesse funzionare, provate con:

```
sudo modprobe sun4i_csi0
```

Adesso per testare il corretto funzionamento del modulo camera è possibile utilizzare il software Mplayer, digitando:

```
mplayer tv: //
```

La libreria API per questo modulo camera è ancora in fase di sviluppo, ma è possibile sviluppare un proprio programma per elaborare i dati della fotocamera.

Esempio pratico

Abbiamo appena visto quanto è semplice interfacciare questo modulo camera al nostro computer embedded e il fatto che le sue caratteristiche lo rendono ideale per essere utilizzato, per esempio, per controllare a distanza un robot rover o un drone, o come videocamera di sorveglianza. Non volendovi lasciare soltanto con un concetto di idea in testa, adesso vedremo come bisogna effettuare la configurazione base per lo streaming video. Per questo test utilizzeremo il programma MJPG-streamer, che permette di trasmettere video in streaming, raggiungibile tramite un semplice browser o un programma per la riproduzione video, come ad esempio VLC o MPLAYER. Per seguire le istruzioni questo paragrafo senza problemi è necessario avere già installato sul Banana Pi un server LAMP, come già spiegato in un articolo precedente. Per installare MJPG-streamer, dopo avere eseguito l'accesso sul terminale come superuser, bisogna digitare nel terminale quanto segue:

```
apt-get install libjpeg8-dev imagemagick libv4l-dev

ln -s /usr/include/linux/videodev2.h /usr/include/linux/videodev.h

cd /home/bananapi/

git clone https://github.com/vfonov/mjpg-streamer

cd mjpgstreamer
```

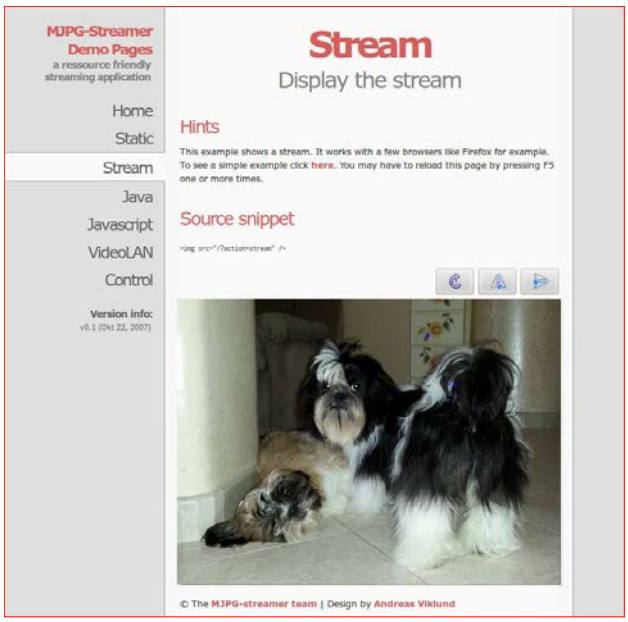


Figura 5: Interfaccia MJPG-STREAM

```
mjpgstreamer# make clean install
```

Se tutto è andato a buon fine, alla fine di queste operazioni dovrebbero apparire delle informazioni simili alle seguenti:

```
install --mode=755 mjpg_streamer /usr/local/bin

install --mode=644 input_uvc.so output_file.so output_udp.so input_file.so /usr/local/lib/

install --mode=755 -d /usr/local/www

install --mode=644 -D www/* /usr/local/www
```

Per compilare e avviare il programma digitiamo:

```
tar xzvf mjpg-streamer.tgz

cd mjpg-streamer

make clean all

export LD_LIBRARY_PATH=.

./mjpg_streamer -o "output_http.so -w ./www"

mjpg_streamer -i "input_uvc.so -y" -o "output_http.so -w /home/bananapi/mjpg-streamer/www/"
```

Nel caso in cui si dovessero riscontrare degli errori, cambiate i permessi della cartella WWW e riprovate. Questa operazione possiamo farla direttamente dal terminale, digitando:

```
chmod 755 www

chmod 644 www
```

Adesso per vedere lo streaming video tramite un web browser basterà andare all'indirizzo <IP_ADDRESS_BANANA_PI>:8080 e muoversi sulla voce streaming, mentre se vogliamo vederlo tramite mplayer, l'indirizzo che dovremo inserire sul programma è <IP_ADDRESS_BANANA_PI>:8080/?action=stream&ignored.mjpg.

Questo presentato è solo un esempio base di applicazione; nel caso in cui doveste decidere di implementare qualcosa di simile nel vostro progetto, dovrete adattarlo in base alle vostre esigenze.

Conclusione

Sono rimasto abbastanza soddisfatto visto che adesso potrò sbizzarrirmi realizzando qualsiasi progetto che mi verrà in mente, potendo implementare una camera che ha tutte le carte in regola per essere definita come la camera adatta ai Makers che amano il Banana Pi. Purtroppo attualmente lo stato di sviluppo del software del modulo camera non è molto avanzato, e questo comporta che bisogna un po arrangiarsi, ma il vero Maker ama complicarsi la vita, quindi questo non è un problema.
Buona sperimentazione!

global spirit

SERVIZI per l'ILLUMINOTECNICA

ITALIAN **FLEXIBILITY**
CHINESE **MANUFACTURE**
EUROPEAN **QUALITY**

PCB

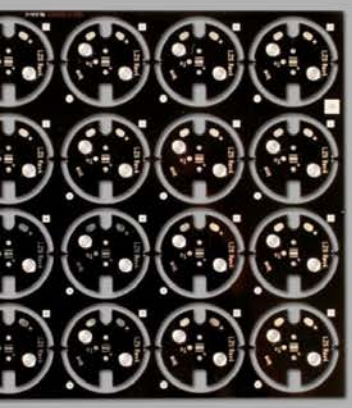
PER
SPOT-LIGHT,
BARRE A LED,
POWER SUPPLY

STAMPI

STAMPI
E STAMPAGGIO,
POWER BOX,
CASE SPOT-LED,
PLASTIC PART

www.ags-one.com

info@ags-one.com



A.G.S. Srl - HEADQUARTERS
Via Capitano Alessio, 181 - 36027 Rosà (VI)
Tel. +39 0424 582757 - Fax +39 0424 582406
P.IVA: 03133060248

R&D DEPARTMENT
Via Tavagnacco, 131 - 33100 Udine (UD) | ITALY
Tel. +39 0432 43254 - Fax +39 0432 487707

A.G.S. China - HEAD OFFICE
Room A, 23Floor, Seaview Plaza No.18, Taizi Road
Shekou District - Shenzhen City - Guang Dong Province
POSTAL CODE: 518067 - Tel. 086-755-82713436

MISURATORE DI RESISTENZA INTERNA BATTERIE (SECONDA PARTE)

di Grazia Ancona

L'utilizzo massivo di dispositivi portatili (cellulari, computer, ecc.) unito ad una forte esigenza di miniaturizzazione hanno portato la tecnologia delle batterie ricaricabili a livelli che solo 20 anni addietro sembravano impensabili. D'altro canto, sono aumentate le esigenze di manutenzione e controllo dei pack batterie stessi. Questo articolo è dedicato a tutti coloro che, per motivi hobbistici o professionali, hanno l'esigenza di verificare in dettaglio lo stato di salute di singoli elementi o di interi pack.

Descrizione del circuito

Prima di passare all'analisi dello schema elettrico sarà bene mostrare le caratteristiche del nostro misuratore:

•Portata: da 1 mΩ a 1,2 Ω;
•Escursione tensione della batteria in test: da 1,2 V a 13,8 V (nominali);
•Temperatura d'esercizio: da 0 a 50° (centigradi);
•Massimo errore di lettura: 4% (tipico 2%);
•Lettura: diretta, su multimetro digitale (portata VDC, 1 mV = 1 mΩ) con almeno 3 cifre decimali ed impedenza non inferiore a 250 K Ω;
•Massima capacità di batteria misurabile con affidabilità: fino a 6 Ah (circa);
•Alimentazione: Pila da 9 V tipo 6LR61 oppure 1604A (entrambe tipo alcalina);
•Indicatore di bassa tensione batteria: a LED (opzionale).

E' molto importante sottolineare che si possono sottoporre al test soltanto batterie con carica pari o superiore all'80% circa. Inoltre, essendo la resistenza interna molto influenzata dalla temperatura, le prove andrebbero fatte nell'ambiente d'esercizio. Quest'ultimo concetto è valido anche quando si vogliono comparare fra loro tipi diversi di accumulatori: questa operazione va fatta con le batterie alle stesse condizioni di temperatura (se necessario, lasciarle acclimatare per qualche minuto). Come si evince leggendo le caratteristiche, il nostro strumento necessita, per la lettura del risultato, di un multimetro digitale con portata in Volt. Questa soluzione, che conferisce al tutto grande semplicità ed economicità, parte dal presupposto che un comune tester digitale è ormai parte del corredo minimo di ogni hobbista elettronico. Inoltre, l'uso della scala a Volt DC, non richiede ulteriori calcoli in quanto la lettura è diretta: 1 mV è pari ad 1 mΩ. Il lettore esperto potrà progettare per suo conto un eventuale visualizzazione su display LCD, previa conversione A/D naturalmente. In questo caso si dovrà anche prevedere un fattore di scala variabile da 1200:1 ma, come ben sappiamo, nulla resiste alla buona volontà… Fatte queste premesse, possiamo passare al dettaglio delle soluzioni circuitali scelte, prima, però,sarà meglio dare un'occhiata allo schema a blocchi. Esso è visibile in figura 9. Il primo blocco a sinistra è un normalissimo oscillatore ad onda quadra che ha il compito di produrre un segnale, avente duty-cicle simmetrico, dal periodo di circa 20 ms. La semionda a livello alto di questo segnale chiuderà ciclicamente l'interruttore del carico a corrente costante (secondo blocco da sinistra) consentendo, alla batteria sotto test, di erogare una corrente ben precisa per tutta la durata dell'impulso (circa 10 ms). Una corrente molto più debole è invece prelevata sul livello basso, si tratta della corrente di stabilizzazione a cui abbiamo già accennato. Il gradino di corrente così ottenuto genera, a causa della resistenza interna della batteria, una segnale alternato avente la stessa geometria della funzione sopra accennata (onda quadra). Tale forma d'onda è prelevata a valle del condensatore di

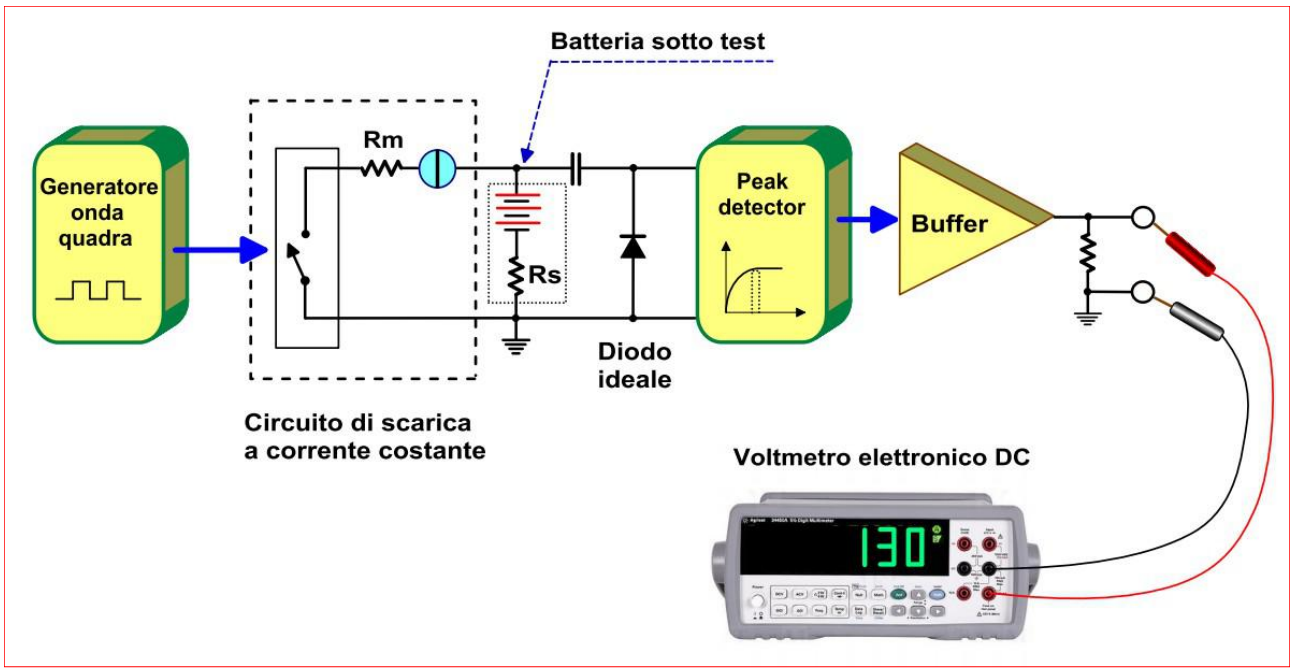


Figura 9: Schema a blocchi

blocco C, esso ha il non trascurabile compito di eliminare la componente continua (tensione di batteria) non interessante ai fini della misura. La presenza di C impone di "fissare" il segnale (ovvero restituirgli la sua intrinseca componente continua) ed a questo compito è chiamato un diodo "ideale". Un normale diodo shottky non avrebbe risolto il problema poiché il suo potenziale di conduzione, a polarizzazione diretta, non sarebbe mai stato inferiore a 0,16 V. Valore, quest'ultimo, ben superiore al mV che noi dovremmo leggere nella misura più sensibile prevista! Il blocco seguente è incaricato di trasformare il segnale ad onda quadra in una componente continua pari al valore di cresta della semionda positiva. In sostanza, un vero e proprio "peak detector" (rivelatore di valore massimo). Possiamo anche considerare questa sezione circuitale una specie di memoria analogica che si "rinfresca" ogni 10 ms. Ultima fatica, chiudere questa tensione su un carico a relativamente bassa impedenza in modo che un eventuale voltmetro, con la sua resistenza interna comunque presente, non disturbi i valori di tensione misurati. Tutto questo si traduce nello schema elettrico di figura 10. Passiamo adesso al dettaglio dei vari blocchi da un punto di vista più specificatamente elettronico. Il generatore di onda quadra è un "sempre verde" NE555 (fare attenzione ad usare la versione C/MOS e non quella TTL!) nella sua configurazione astabile. Per ottenere il duty-cicle al 50% si è reso necessario l'impiego del diodo D1. Il segnale uscente è applicato a due partitori: che hanno il compito di ridurre la semionda positiva a 110 mV e di mantenere il potenziale di quella negativa a 10 mV circa. Questi due valori di tensione, una volta applicati al generatore di corrente costante formato da U2 e Q1, serviranno a prelevare un gradino di corrente alla batteria sotto test. In particolare, dato il valore di R7 pari a 0,1 Ohm, il primo genererà una corrente pari a 1,1 A (livello alto) mentre il secondo si accontenterà di una corrente più modesta: 0,1 A (livello basso). Mantenere un "piedistallo" di corrente durante la parte bassa del segnale di pilotaggio ha lo scopo di fornire, come ampiamente accennato, una corrente di stabilizzazione necessaria per una misura accurata. Dovendo essere l'ampiezza della corrente prelevata in fase di misura pari a d 1 A, condizione necessaria per mantenere una perfetta equivalenza Volt-Ohm in fase di lettura, occorrerà aggiungere a tale valore i 100 mA sopra citati. Il gruppo RC formato da R5 e C4 e posti all'uscita di U2, ha il compito di rendere più stabile

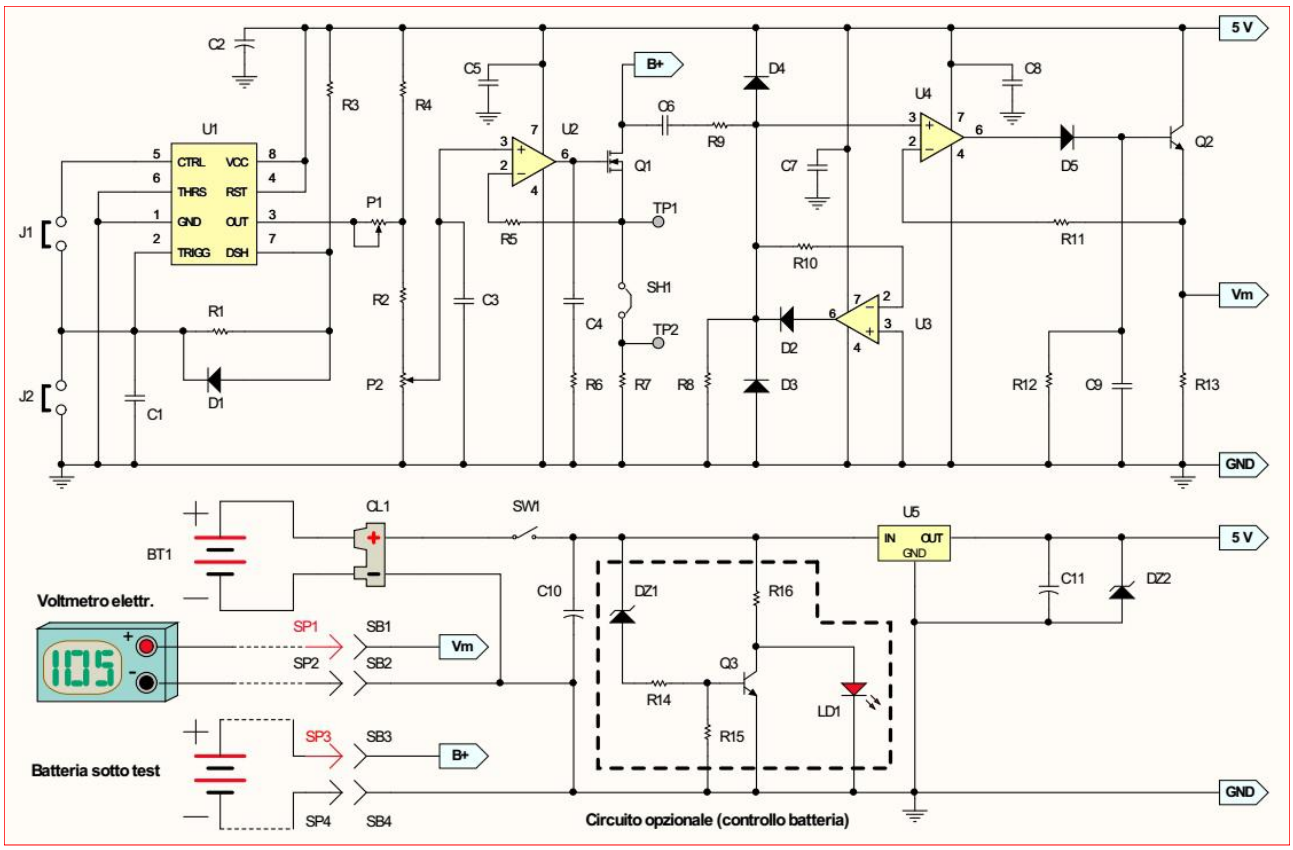


Figura 10: Schema elettrico

l'operazionale e di escludere eventuali auto-oscillazioni. Passiamo ora al diodo "ideale", esso si serve dell'operazionale U3 configurato in forma canonica per questo tipo d'applicazione. Con tale schematizzazione, il diodo D2, quando interessato da polarizzazione diretta, conduce anche con segnali di pochi µV. I diodi D3 e D4 hanno invece semplici funzioni di protezione. Essi devono evitare che alti valori di potenziale vadano a chiudersi sugli ingressi degli operazionali nell'istante in cui colleghiamo la batteria sotto test. In figura 11 possiamo osservare la forma d'onda attestata sul pin 3 di IC3 (ingresso non invertente) durante una misura. Si noti come il segnale, essendo in regime di scarica capacitiva, mostri un certo abbassamento durante il livello alto. Questo fattore è del tutto ininfluente ai fini della misura. Anche il circuito del "peak-detector" è svolto in maniera canonica, unica particolarità richiesta: il condensatore C9 deve essere di ottima qualità e possibilmente in tecnologia al policarbonato. Un normale transistor 2N2222 (anche il PN2222, equivalente plastico, va bene lo stesso) realizza un buffer di corrente ad alta stabilità la cui reazione in tensione è applicata direttamente al pin invertente del nostro peak-detector". In questo modo possiamo prelevare la tensione da misurare su un carico a impedenza relativamente bassa, questa soluzione è necessaria per non rischiare di falsare la misura di un voltmetro elettronico ad alta impedenza. L'alimentazione è stabilizzata da un regolatore a 3 terminali tipo L7805 mentre un banale circuito a soglia permette l'accensione di un LED non appena la batteria scende sotto 8,2 Volt circa. Questo circuito, tratteggiato nello schema elettrico di figura 10, è assolutamente opzionale. Nel caso si decida di escluderlo, non andranno reperiti i componenti segnati da un asterisco nel part-list.

Note costruttive e taratura

Essendo la nostra realizzazione uno strumento, tutti i resistori saranno a strato metallico con tolleranza massima al 2% mentre, per le capacità si dovranno scegliere tecnologie ad alta stabilità

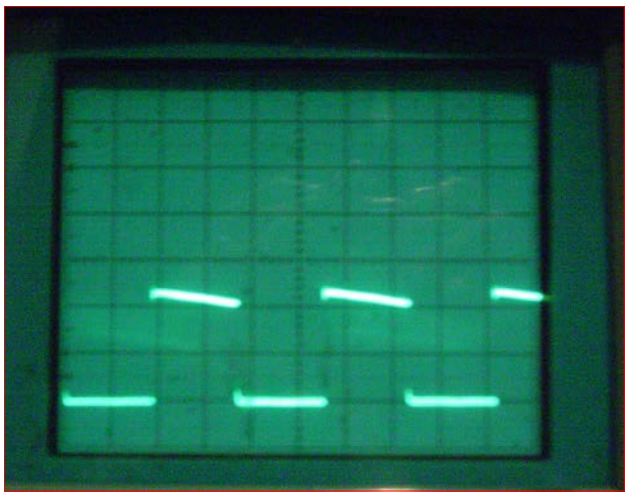


Figura 11: Segnale sul pin 3 di U4 (time-base 5 ms, 20 mV/cm)

(poliestere o propilene). Fanno eccezione C5, C7 e C8 che, essendo shunt, dovranno essere ceramici o multistrato ed andranno montati molto vicino ai terminali di alimentazione di ogni operazionale. Questi ultimi sono stati scelti per le loro eccellenti caratteristiche (rail-to-rail, bassissima deriva, ecc.) e costo decisamente basso ma sono prodotti soltanto dalla BURR-BROWN. L'alimentazione potrà essere fornita anche da batterie ricaricabili al NiCd ma, in questo caso, ne occorreranno almeno 7 da 1,2 Volt e DZ1 dovrà essere un BZX84C6V2 per consentire un avviso di batteria scarica a tensione più bassa (circa 7,4 Volt). R7 merita una nota a parte. Per avere un elemento ad alta stabilità sarà meglio disporre di 10 resistori a strato metallico da 1 Ohm ¼ di W posti in parallelo. Infatti, le normali resistenze a filo da 1 W, possono essere termicamente "rumorose" e disturbare le letture di resistenze serie nell'ordine dei pochi mΩ. Il MOSFET Q1 deve essere necessariamente montato su dissipatore (è sconsigliabile interporre un isolante fra il DRAIN e il dissipatore vero e proprio, meglio tenere isolato quest'ultimo dal resto del circuito) tenendo conto che, alla massima potenza ammissibile dallo strumento, esso dovrà dissipare una potenza pari al 50% di 13,8V x 1,1A ovvero circa 8,5 W (corrente di stabilizzazione compresa). Naturalmente il test avrà luogo in pochi secondi ma, considerata la dissipazione termica non trascurabile, sarà necessaria una discreta superficie di raffreddamento pari ad almeno 50 cmq (se priva di alettatura). Con alettature molto elaborate questa superficie può ridursi oppure, in caso di alimentazione con batterie ricaricabili, potremmo inserire una ventolina miniatura. Per la taratura sono necessari i seguenti componenti e strumenti:

•Un Amperometro DC (possibilmente digitale);
•Un voltmetro digitale DC;
•Un alimentatore con uscita pari almeno a 6 V e corrente 2 A (in alternativa si può usare una batteria efficiente con capacità non inferiore a 6 Ah);
•Un resistore da 1 W (minimo) di valore compreso fra 0,1 e 0,15 Ohm;
•Una batteria da misurare avente resistenza interna sconosciuta.

Effettivamente, al posto dell'amperometro, potremo utilizzare un voltmetro in DC ma, dato che poi la misura sarebbe ai capi di R7, rimarremmo vincolati alla precisione di questa per ottenere l'accuratezza necessaria. Ovviamente, il costruttore sprovvisto di amperometro dovrà necessariamente ricorrere a quest'ultima metodica. Procedere come segue:

1.Collegare l'alimentazione ed attendere per circa 2 minuti;

2.Mettere in corto il ponticello J2 e leggere, con un voltmetro DC, la tensione su TP1 e massa (vedi schema elettrico). Regolando P2 essa dovrà essere portata a circa 110 mV. Aprire quindi J2;

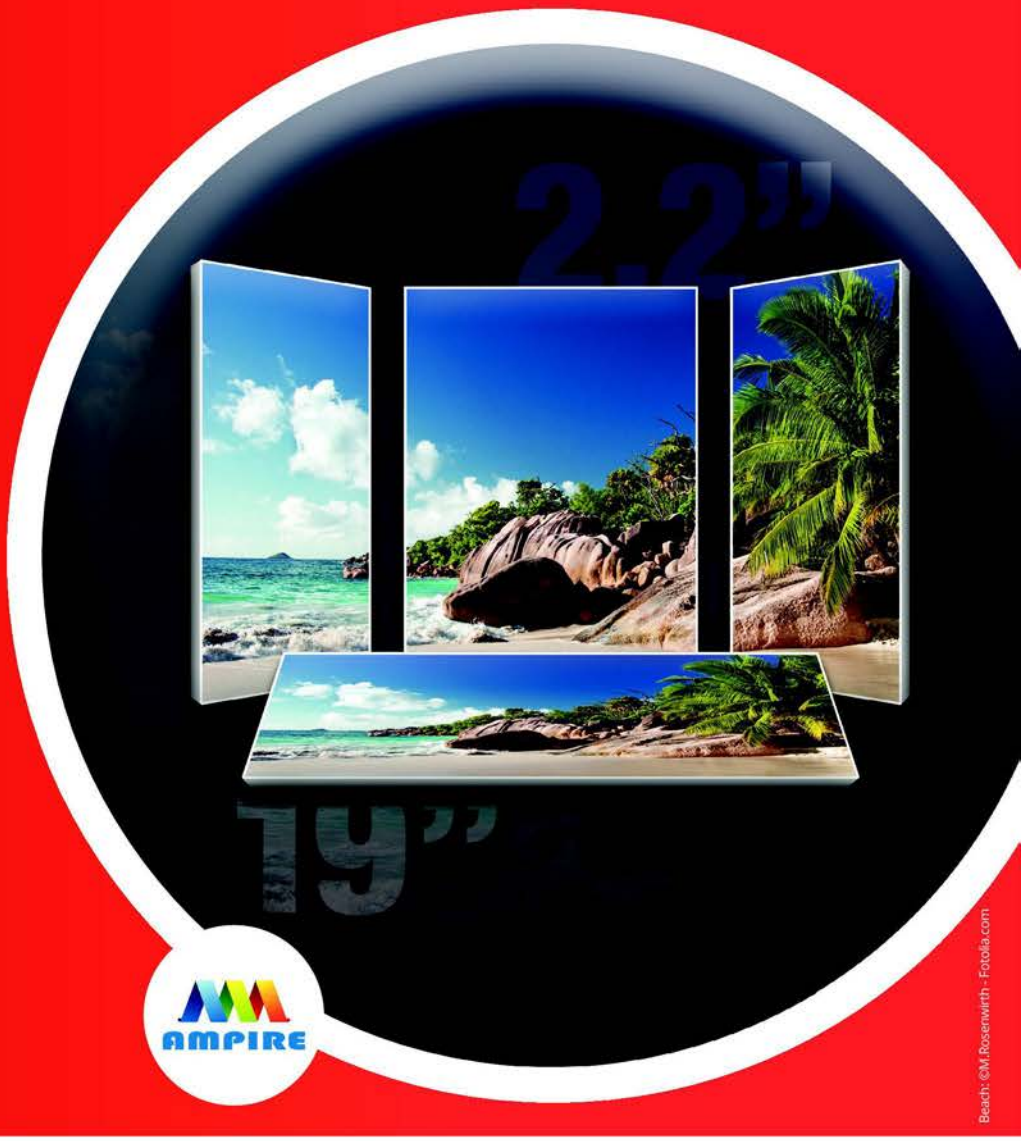
3.Mettere in corto il ponticello J1 e leggere, con un voltmetro DC, la tensione su TP1 e massa (vedi schema elettrico). Regolando P1 essa dovrà essere portata a circa 10 mV. Scollegare quindi l'alimentazione. Aprire quindi J1;

4.Collegare un alimentatore da banco al posto della batteria da misurare (attenzione

CODICO®

IPS (In Plane Switching) su tutti modelli di TFT!

- Da 2.2" a 19"
- Risoluzione da 240x320 a 1920x1200
- Contrasto 900:1
- Angolo di visione 85° da ogni direzione



AMPIRE

Beach: ©M.Rosenwirth - Fotolia.com

Contatto: +39 0362 364358 | sergio.rossi@codico.com | www.codico.com/shop

Componente	Valore	Quantità	Descrizione
BT1	Batt. 9 V	1	Tipo 6LR61 oppure 1604A (Alkaline)
C1, C6	1 uF 50 VL	2	Poliestere o policarbonato
C2, C10, C11	10 uF 25 VL	3	Cond. Elettrolitico
C3	10 nF 50 VL	1	Ceramico
C4	33 nF 50 VL	1	Ceramico o poliestere
C5, C7, C8	100 nF 50 VL	3	Multistrato (va bene anche ceramico)
C9	2.2 uF	1	Policarbonato
CL1	-	1	Clips per batteria a 9 V
D1, D2, D3, D4, D5	1N4148	5	Sostituibile con 1N4149
DZ1*	BZX78C6V8	1	Zener
DZ2	BZX78C6V8	1	
J1, J2	-	2	Jumper normalmente aperto
LD1*	LED rosso	1	Diametro 3 mm
P1	470 Ohm	1	Trimmer lineare
P2	10 Kohm	1	Trimmer multigiri
Q1	IRF540	1	MOSFET canale N (TO-220)
Q2	2N2222	1	Transistor NPN (sostituibile con PN2222)
Q3*	BC547	1	Transistor NPN
R1	15 Kohm	1	
R2	220 Kohm	1	
R3	12 Kohm	1	1/4 watt, 2%, a strato metallico
R4	2,2 Kohm	1	
R16*	2,2 Kohm	1	
R5, R10, R11, R13	10 Kohm	4	
R14*	10 Kohm	1	
R6	33 ohm	1	
R7	0,1 ohm	1	2W, res. a filo oppure vedi testo
R8	68 Kohm	1	1/4 watt, 2%, a strato metallico
R9	1 Kohm	1	
R12	1 Mohm	1	
R15*	3,3 Kohm	1	
SB1, SB3	-	2	Boccola rossa femmina diam. 3 mm
SB2, SB4	-	2	Boccola nera femmina diam. 3 mm
SP1, SP3	-	2	Spinotto rosso diam. 3 mm
SP2, SP4	-	2	Spinotto nero diam. 3 mm
SH1	-	1	Jumper normalmente chiuso
SW1	-	1	Interruttore unipolare
TP1, TP2	-	2	Test point (opzionali)
U1	NE555C	1	Timer di precisione tipo C/MOS (DIP-8))
U2, U3, U4	OPA344	3	Amplificatore operazionale (DIP-8)
U5	78L05	1	Regolatore di tensione 3 terminali (TO-92)

* Vedi testo (opzionali)

Elenco componenti

alla polarità!). Esso deve fornire una tensione compresa fra 3 e 6 VDC e limitatore di corrente, se presente, regolato a 2 A minimo;

5. Aprire il ponticello SH1 (vedi schema elettrico) ed inserire un amperometro. Fra TP1 e TP2 (positivo su TP1). Ricollegare l'alimentazione e chiudere il ponticello J2. Regolare finemente P2 fino a leggere una corrente pari a 1100 mA (oppure 110 mV DC ai capi di R7). Aprire quindi il ponticello J2;

6.Mettere in corto il ponticello J1 e regolare finemente P1 fino a leggere una corrente pari a 100 mA (oppure 10 mV DC ai capi di R7). Aprire quindi il ponticello J1;

7.Ripetendo la prova al punto 5), controllare nuovamente se la corrente è effettivamente pari a 1100 mA. In caso contrario ritoccare leggermente P2;

8.Scollegare quindi l'alimentazione e l'alimentatore da banco dai punti di misura. Chiudere il ponticello SH1 ed aprire i ponticelli J1 e J2 se ancora chiusi.

A questo punto la taratura vera e propria è ultimata, occorre solo verificare se le misure prodotte sono esatte. A tal scopo occorrerà un accumulatore avente resistenza interna ignota che sottoporremmo ad una misura annotandone quindi il valore. Chiameremo quest'ultimo Rx. Adesso poniamo in serie al nostro accumulatore la resistenza di valore noto compresa fra 0,1 e 0,15 Ohm, che chiameremo Rr, e ripetiamo la misura. Il valore letto sarà, chiamato Rtot sarà pari a:

$$Rtot = Rx + Rr$$

Essendo Rn nota, sarà facile verificare l'esattezza della misura. Facciamo un esempio pratico: Supponiamo che la misura della batteria abbia fornito 55 mV, ovvero 55 mΩ. La seconda misura, con Rr in serie alla batteria e con valore pari a 0,12 Ω , visualizza 177 mV, ovvero 177 mΩ.. Facendo un rapido calcolo:

$$Rx = Rtot - Rr$$

ovvero:

$$177 - 120 = 57$$

Valore che, a meno di un trascurabile errore del 3,5%, ricalca quello ottenuto con la batteria senza la resistenza aggiuntiva (55 mΩ).

Conclusione

Era mia intenzione presentare un progetto "aperto" che potesse, nei limiti stabiliti dalla fantasia e competenza del lettore, evolvere migliorando prestazioni e performance. Sebbene le norme di taratura possano sembrare un pochino troppo precise, vi garantisco che il tempo speso per semplificare processi e progetto è stato il massimo possibile. Lascio a voi il giudizio se questo obiettivo è stato colto in tutto o, almeno, in parte! Rimango come sempre a vostra disposizione per chiarimenti e consigli.

Bibliografia

Handbook of batteries di D. Linden e T. B. Reddy (McGraw-Hill)
Comparison of several methods for determining internal resistance of Lithium Ion Cells
(Hans – Georg Schweiger ed altri)

Tecnologia costruttiva	Capacità in Ah	Resistenza interna a 20° (espressa in mΩ)	Note
Ni-Cd	0,55	120 - 170	La misura dipende molto dalla qualità dell'accumulatore
	Tipo "D" (1,2 Ah)	40 - 80	
Ni-MH	0,5	650 - 780	Il pack potrebbe contenere elementi di protezione (fusibili, PTC, ecc.) che alterano la misura di resistenza interna
Li-ion	0,5	300 - 400	
Pb-acido	6	20 - 30	Misura effettuata al 90% di carica
Alcalina	Tipo "AA" (1,5 Volt)	140 - 170	Non ricaricabili
	6LRS61 (9 Volt)	1000 - 2000	
Zinco-Carbone	6LRS61 (9 Volt)	25000 - 30000	Non ricaricabile

Tabella 1: Esempi di resistenza interna (Rs) espressi per tecnologia e tipo di batteria

RICEZIONE E DECODIFICA DEL DCF77 UTILIZZANDO IL TIMER1 DEL PIC16F887 DELLA MICROCHIP

di Alberto Trasimeni

La stazione DCF77, è gestita dal Physikalisch-Technische Bundesanstalt,e largamente usata in Italia. La sigla significa: D=Deutschland (Germania), C=segnale ad onde lunghe, F=Francoforte, 77=Frequenza di 77,5 kHz.Il segnale orario trasmesso è dovuto ad un orologio atomico (al cesio) che si trova a Mainfingen distante circa 25Km da Francoforte. Il segnale è ben ricevibile a una distanza di 500 km, ma con una certa intermittenza (che dipende dalle caratteristiche atmosferiche di propagazione) e può essere ricevuto ad una distanza massima di 1000 km. Se la distanza risulta essere superiore a 1000 km la sincronizzazione è difficile per non dire sporadica. Ad esempio nella città di Roma un buon ricevitore riesce a sincronizzarsi sempre di notte e qualche volta anche di giorno, per questo vengono usati due trasmettitori indipendenti da 50 kW ciascuno per consentire la continuità del servizio.

La realizzazione di un orologio basato sul DCF77 pone non poche problematiche al progettista, che decide di scrivere partendo da zero il proprio software di gestione del segnale. Pur avendo a disposizione un buon ricevitore fornitomi dalla ditta ECAT di Mondovì è difficile ottenere una ricezione ottimale durante il giorno, per questo ho usato un simulatore che mi ha permesso di sviluppare l'algoritmo di ricezione in modo semplice. Il simulatore prevede di fissare inizialmente a piacere l'ora e i minuti da cui far partire la trama di trasmissione. Tale orario viene successivamente incrementato ed aggiornato ogni minuto, inoltre è possibile introdurre dei disturbi nel segnale trasmesso per renderlo più credibile e al contempo verificare la qualità dell'algoritmo. Tutto ciò mi ha consentito con piccole correzioni di validare il programma caricato nel microcontrollore. La figura 1 sotto riportata mette in evidenza la copertura del segnale DCF77.

Figura 1; Segnale DCF77

La struttura del segnale DCF77

La trasmissione del segnale, come precedentemente detto, si propaga agevolmente soprattutto di notte, ciò è dovuto al contributo della stratosfera che si addensa in assenza della radiazione solare. La modulazione usata è del tipo AM e la portante viene interrotta per 100 ms o 200 ms all'inizio di ogni secondo, tali interruzioni rappresentano rispettivamente il bit trasmesso. Vale a dire se l'interruzione è di 100 ms si sta trasmettendo il bit "0" mentre se l'interruzione dura 200 ms si sta trasmettendo il bit "1".In sostanza in uscita dal ricevitore avremo quindi un "1" che dura 100 ms per rappresentare lo "0" oppure un "1" che dura 200ms per rappresentare un "1". Tali tempi sono del tutto teorici, in quanto come vedremo più avanti oscillano con una variazione +/- del 10% rispetto ai valori

DCF77 time code

Bit	Weight PM	Weight AM	Meaning
:00	1	M	Start of minute. Always 0.
:01	1		Civil warning bits,[16] provided by the *Bundesamt für Bevölkerungsschutz und Katastrophenwarnung* (Federal Office of Civil protection and Disaster Relief). Also contains weather broadcasts.[15][17]
:02	1		
:03	1		
:04	1		
:05	1		
:06	1		
:07	1		
:08	1		
:09	1		
:10	0		
:11	0		
:12	0		
:13	0		
:14	0		
:15	R		Call bit: abnormal transmitter operation.[15] Previously: backup antenna in use.
:16	A1		Summer time announcement. Set during hour before change.
:17	Z1		Set to 1 when CEST is in effect.
:18	Z2		Set to 1 when CET is in effect.
:19	A2		Leap second announcement. Set during hour before leap second.

Bit	Weight PM	Weight AM	Meaning
:20	S		Start of encoded time. Always 1.
:21	1		Minutes 00–59
:22	2		
:23	4		
:24	8		
:25	10		
:26	20		
:27	40		
:28	P1		Even parity over minute bits 21–28.
:29	1		Hours 0–23
:30	2		
:31	4		
:32	8		
:33	10		
:34	20		
:35	P2		Even parity over hour bits 29–35.
:36	1		Day of month. 01–31
:37	2		
:38	4		
:39	8		

Bit	Weight PM	Weight AM	Meaning
:40	10		Day of month (continued)
:41	20		
:42	1		Day of week Monday=1, Sunday=7
:43	2		
:44	4		
:45	1		Month number 01–12
:46	2		
:47	4		
:48	8		
:49	10		
:50	1		Year within century 00–99
:51	2		
:52	4		
:53	8		
:54	10		
:55	20		
:56	40		
:57	80		
:58	P3		Even parity over date bits 36–58.
:59	0		Minute mark: no amplitude modulation.

Figura 2; Struttura del segnale DCF77

nominali, in relazione alle condizioni atmosferiche e alla distanza di ricezione. Da quanto ora affermato appare evidente che per decodificare il segnale, è necessario realizzare un discriminatore temporale. Le soluzioni possibili possono essere due, di tipo hardware o software; quella da me realizzata è una soluzione mista, infatti utilizzerò un semplice multivibratore monostabile "retriggerabile", per stabilire l'inizio della trama di trasmissione, che avviene ogni minuto, delegando al timer1 del PIC16F887 il compito di realizzare il discriminatore temporale, e quindi provvedere alla decodifica del segnale. Tutta l'informazione trasmessa dalla stazione DCF77 avviene in 58 impulsi (uno al secondo), il 59° impulso è soppresso ad indicare che sta iniziando la trama del minuto successivo. Come mostra la figura 2, tutta l'informazione trasmessa è organizzata a gruppi di bit, e ciascun gruppo trasporta una diversa informazione. Ad esempio, i minuti sono trasmessi dall'impulso 21 all'impulso 29, compreso il bit di parità pari, di cui è necessario tenerne conto per il controllo dell'informazione ricevuta. Un'osservazione è quella relativa al bit 16 che ci dice che nell'ora successiva a quella di ricezione, entrerà in funzione l'ora solare/legale, ma tale informazione in effetti è già contenuta nella trama dell'orario perché aggiornata automaticamente. Accennavo precedentemente al bit di parità, questo si troverà ad "1", se il numero dei bit ricevuti pari ad "1", ad esempio nella ricezione dei minuti(escluso il bit di parità), è un numero dispari, in modo tale che il numero dei bit ad "1" ricevuti in quel

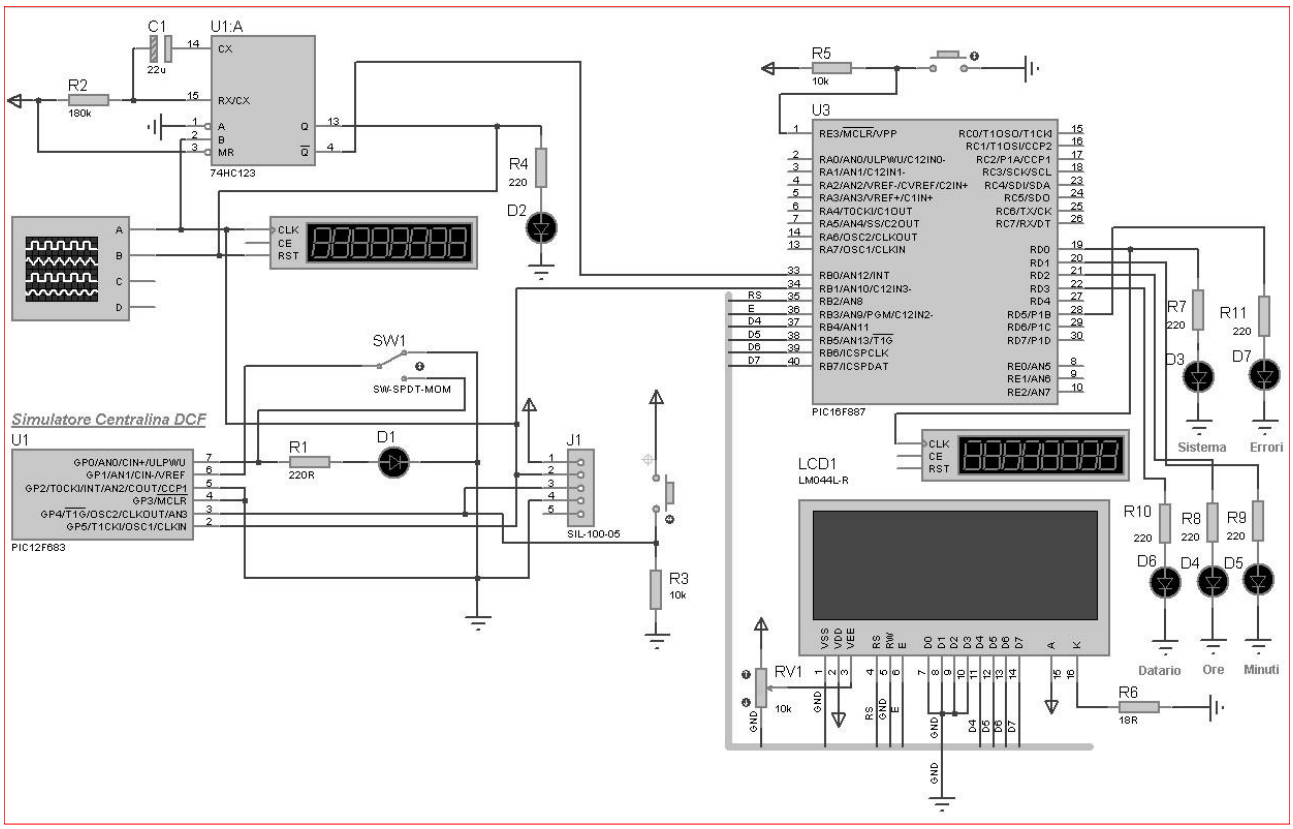


Figura 3: Schema elettrico

pezzo di trama risulti sempre pari. I successivi bit di parità si troveranno all'impulso 35 per quanto riguarda le ore e all'impulso 58 per quanto riguarda il datario. Pur avendo a disposizione il bit di parità per controllare le informazioni ricevute, voglio sottolineare che ricevendo, ad esempio, i minuti in modo errato, questo errore può essere compensato da una ricezione non corretta del relativo bit di parità. In questo caso accetteremmo una informazione che risulta errata, dovuta essenzialmente alla fragilità della trasmissione come accennato in precedenza. Quindi se si vuole realizzare un orologio statisticamente esatto è bene affiancare al DCF77 un RTC (Real Time Clock) come ad esempio DS1307, che ci aiuti a stabilire se effettivamente la ricezione delle informazioni è avvenuta in modo corretto. Quando ciò avviene è possibile caricare l'informazione relativa all'orario al datario del DCF77 nell'RTC. Lo schema di principio del sistema da me realizzato per la ricezione della trasmissione è riportato nella figura sottostante, figura 3. Nella

Tabella 1: Funzione dei diodi Led

LED	SIGNIFICATO	STATO ATTIVO
D2	Segnalazione fine trama	OFF
D3	Attività del sistema	TOGGLE
D4	Segnalazione ricezione ore	ON
D5	Segnalazione ricezione minuti	ON
D6	Segnalazione ricezione datario	ON
D7	Segnalazione ricezione errori	ON

Wherever You Are – EBV IoT Solutions from NXP

The EBV IoT
Smart, Secure, Connected – Everywhere

Primo dispositivo ad avvalersi dei core ARM Cortex-A9 e ARM Cortex-M4, il processore per applicazioni di NXP, i.MX 6SoloX, offre una soluzione multi-mercato altamente integrata. Grazie ad un'implementazione sicura e robusta, rende possibile l'esecuzione simultanea di più ambienti software, per portare sicurezza e connettività nelle case e nelle auto, nell'ambito dell'Internet delle Cose (IoT). Il processore i.MX 6SoloX garantisce la massima efficacia e sicurezza durante il partizionamento delle risorse di sistema, quali memoria e periferiche, ed è supportato da un solido ecosistema che consente di realizzare migliori soluzioni personalizzate e di ridurre il time-to-market.

Per il know-how e il supporto applicativo, contatta il tuo partner locale di EBV, distributore leader di semiconduttori nella regione EMEA specializzato nei prodotti NXP, e visita anche ebv.com/imx.

Distribution is today. Tomorrow is EBV!

www.ebv.com/it

NXP | EBV Elektronik
I An Avnet Company I

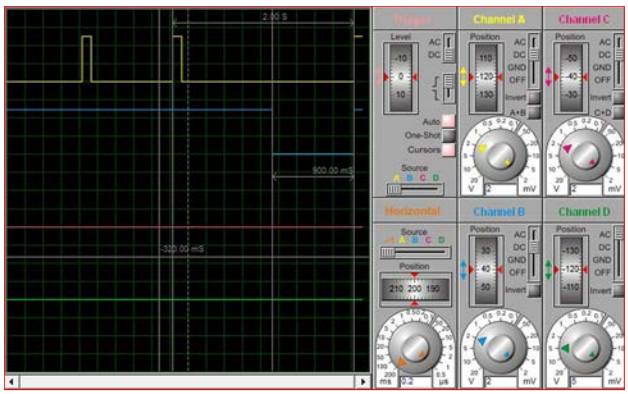


Figura 4: Segnale di fine trama

realizzazione pratica, come mostrerò successivamente, il simulatore è stato sostituito con il ricevitore della ditta ECAT di Mondovì. Il circuito che consente la sincronizzazione con il trasmettitore è costituito dal multivibratore monostabile retriggerabile 74HC123. Il cui funzionamento è assai semplice: portando il pin1 (A) al potenziale di massa e il pin3 (MR) a Vcc ed applicando al pin2 (B) gli impulsi provenienti dalla centralina (un impulso ogni secondo) l'uscita pin4 (QN) resta sempre bassa fintanto che gli impulsi sono presenti. Quando si raggiunge il fine trama tra l'ultimo impulso della trama attuale e il primo della successiva intercorre un tempo di due secondi, in queste condizioni (il tempo di caduta del monostabile è di 1.1 sec) l'uscita QN si porta a livello logico "1" generando un interrupt (INTE) di inizio ricezione. Ma consideriamo anche il caso che dopo un l'interrupt di inizio ricezione, un impulso dovesse essere assente per più di 1 secondo (prima della fine della trama), si genererà un secondo interrupt, segnalando al microcontrollore che la trasmissione si è interrotta prima della fine, originando così un errore di fine trama. Successivamente il microcontrollore provvederà automaticamente alla inizializzazione di una nuova ricezione che partirà dopo aver ricevuto il nuovo fine trama da parte del trasmettitore. La resistenza R2 e la capacità C1, che determinano il tempo di caduta del monostabile, sono facilmente calcolabili mediante la relazione:

Tw =0,28*R2*C1*(1+0,7/R2)

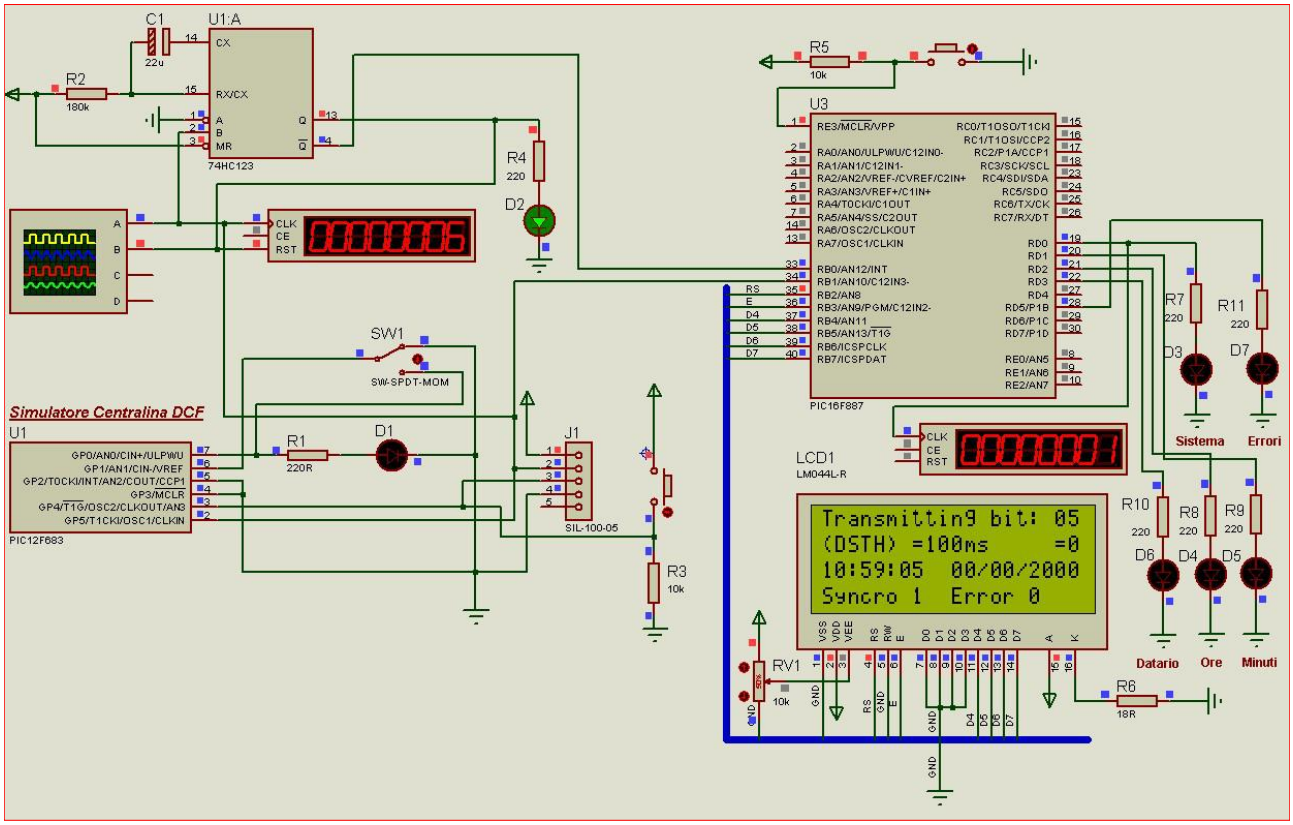


Figura 5: Simulazione del sistema con Proteus

I valori determinati sono: R2=180k e C2=22uf, la resistenza dovrà essere di precisione con tolleranza almeno del 5%, mentre il condensatore dovrà essere del tipo al tantalio. I led presenti nello schema hanno il seguente significato come da tabella 1. Prima di illustrare il software di sistema riporto nelle figure 4 e 5 l'andamento del segnale di fine trama e della ricezione del DCF77 utilizzando il simulatore Proteus. Osservando la figura precedente può sembrare che la ricezione del datario sia errata, ma non è così, in quanto nel software del simulatore non è stata ancora implementata la trasmissione del datario. Data la lunghezza del programma illustrerò solo le parti fondamentali e cioè:

1) La configurazione del timer1 in modo che funga da discriminatore temporale;
2) La gestione del timer1 per il corretto riconoscimento dei bit trasmessi;

```
Elenco componenti

R2        180 KΩ  1/4 W  5%
C1        22µF  25 V tantalio
U3        PIC16F887
U1:A      74HC123/CD74HC123
CENTRAL
DCF77     ECAT
BreadBoard        WISH consigl.
```

3) La gestione dell'interruzione di inizio ricezione.

Il software e la simulazione con Proteus saranno messi a disposizione del lettore e scaricabili dal sito dell'editore, per chi volesse cimentarsi nella realizzazione del progetto. Il programma è stato scritto in MikroC (rel. 6.6.2) e la board di sviluppo utilizzata è la EasyPic6. Voglio anche precisare che si può impiegare un ricevitore diverso da quello da me utilizzato, ad esempio quello rintracciabile online e fornito da Conrad, economico e ben funzionante anche con bassi segnali.

Listato 1

```
// Timer1 Registers:
// Prescaler=1:8; TMR1 Preset=0; Freq=3,8147Hz; Period=262.144.000 ns
T1CON.T1CKPS1 = 1; // bits 5-4  Prescaler Rate Select bits
T1CON.T1CKPS0 = 1;
T1CON.T1OSCEN = 1; // bit 3 Timer1 Oscillator Enable Control: bit 1=on
T1CON.T1SYNC  = 1; // bit 2 Timer1 External Clock Input Synchronization Control bit: 1=Do
not synchronize external clock input
T1CON.TMR1CS  = 0; // bit 1 Timer1 Clock Source Select bit: 0=Internal clock (FOSC/4) / 1
= External clock from pin T1CKI (on the rising edge)
T1CON.TMR1ON  = 0; // bit 0 enables timer
TMR1H = 0x00;      // preset for timer1 MSB register
TMR1L = 0x00;      // preset for timer1 LSB register
```

Listato 2

```
#define  MAXZERO  30000     // tempo massimo zero 120ms
#define  MINZERO  18750     // tempo minimo  zero 75ms

#define  MAXUNO   55000     // tempo massimo uno  220ms
#define  MINUNO   42500     // tempo minimo  uno  170ms
```

Vuoi semplificare la progettazione e ridurre il sovraccarico dei microcontroller?

I DAC con EEPROM integrate mantengono le impostazioni allo spegnimento

I Digital-To-Analog Converter (DAC) MCP48FEBXX di Microchip, con EEPROM integrate, mantengono le impostazioni allo spegnimento per eliminare la necessità di memoria esterna oltre a ridurre il sovraccarico del microcontroller.

Disponibili in versione single o dual channel con Serial Peripheral Interface (SPI), la famiglia offre diverse modalità di shut-down per ridurre drasticamente il consumo di potenza, ed è ideale per applicazioni alimentate a batterie. Inoltre, questi dispositivi si contraddistinguono anche per un basso errore di Differential Nonlinearity (DNL) per sostenere output monotono e ridotto errore Integral Nonlinearity (INL), per una migliore linearità.

Lo stesso set di funzionalità, ma senza la EEPROM integrata, viene offerto dalla famiglia MCP48FVBXX come alternativa più economica, per applicazioni che non richiedano memoria.

microchip DIRECT
www.microchipdirect.com

MICROCHIP

www.microchip.com/get/eumcp48fxbxx

Il nome e logo Microchip è un marchio industriale registrato di Microchip Technology Incorporated negli U.S.A. e altri Stati. Tutti gli altri marchi industriali menzionati nel presente documento appartengono ai rispettivi titolari
© 2016 Microchip Technology Inc. Tutti i diritti riservati. DS20005499A. MEC2052Ita04/16

Listato 3

```
//---- Interrupt RB1 di lettura dato ----------//
   if(INTCON.RBIF==1)
    {
     if(PORTB.B1==1) // se riceve fronte salita inpulso dalla centralina
      {
        if(flag_finetra==1)
        {
          *Timer1=0x0000;
          T1CON.TMR1ON  = 1;    // abilito il timer
          if(flag_impulsi==0)
           {
             flag_impulsi=1;
           }
          else
           {
            impulsi+=1;
           }
        }
      }
     if(PORTB.B1==0)           // se riceve fronte discesa inpulso
      {
        if(flag_finetra==1)
         {
           T1CON.TMR1ON  = 0;              // disabilito timer1
           scatti_timer1=(TMR1H<<8)+TMR1L; // leggo il timer
           if((*Timer1>=MINZERO)&&(*Timer1<=MAXZERO))
            {
              valbit = 0x00;
            }
           if((*Timer1>=MINUNO)&&(*Timer1<=MAXUNO))
            {
              valbit = 0x01;
            }
```

Configurazione del timer1

Prima di configurare il Timer1 del PIC definiamo un parametro che chiameremo DSTH e che misurerà la durata dell'impulso inviato dal trasmettitore che ci permetterà successivamente di discriminare tra bit "0" e bit "1" inviato. Per effetto sia delle condizioni meteo che della distanza di ricezione imporremo che tale parametro possa variare nei seguenti intervalli:

75ms<= DSTH <=120ms (bit "0")

170ms<= DSTH <=220ms (bit "1")

Per essere sicuri di non commettere errori imporremo che il timeout per il timer1 risulti maggiore del tempo massimo precedentemente definito, in modo particolare imporremo che tale tempo sia di circa 262ms. La relazione che lega il timeout del timer alla frequenza di clock dell'oscillatore interno (8Mhz) e al fattore di divisione del prescaler, è data da:

$$\mathbf{Tout=(Div*4/fosc)*[2^{16}-Preset_timer]}$$

Imponendo: per fosc=8Mhz, Div=8 (fattore di divisione del prescaler), Preset_timer=0 a conti fatti il timeout del timer1 risulterà, Tout=262,14ms. La configurazione per il timer è riportata nel listato 1.

Gestione del timer1

Per la determinazione del DSTH utilizzeremo l'interrupt scatenato dal fronte di salita e di discesa sul pin RB1 del microcontrollore,infatti su tale pin

Listato 4

```
//-------- interrupt RB0 di fine trama ----------//
  if(INTCON.INTF==1)
  {
    if(impulsi==0)
     {
       flag_finetra=1;    // setta il flag per iniziare ricezione
       flag_impulsi=0;    // setta il flag per conteggio inpulsi da zero
       minuti_ric=0;      // cancella campo minuti ricevuti da DCF
       ore_ric=0;         // cancella campo ore ricevute da DCF
       giorno_setti_ric=0;giorno_ric=0;
       mese_ric=0;anno_ric=0;
       bitauno=0;         // cancella campo conteggio 1 ricevuti
       impulsi=0;         // cancella campo inpulsi
       errore_trama=0;errore=0;
       syncro=1;
     }
    else
     {
       flag_finetra=0;    // resetta il flag per iniziare ricezione
       flag_impulsi=0;    // resetta il flag per conteggio inpulsi da zero
       minuti=0; ore=0;giorno_setti=0;mese=0;
       anno=0;minuti_ric=0; // cancella campo minuti ricevuti da DCF
       ore_ric=0;           // cancella campo ore ricevute da DCF
       giorno_setti_ric=0;mese_ric=0;
       anno_ric=0;bitauno=0; // cancella campo conteggio 1 ricevuti
       errore_minuti=0; errore_ore=0;
       errore_datario=0;impulsi=0;   // cancella campo inpulsi
       valbit=0;bitval=0;
       PORTD.B1=0; PORTD.B2=0;
       PORTD.B3=0;errore_trama=1;
     }
    INTCON.INTF=0;
   }
```

viene applicato il segnale proveniente dalla centralina. Se su RB1 è presente un fronte di salita resetteremo il timer1 e lo attiveremo in modo da poter cominciare il conteggio del tempo. Quando arriva il fronte di discesa fermeremo il timer e gli scatti effettuati determineranno la durata dell'impulso. Il numero degli scatti fatti dal timer1 per rispettare gli intervalli del DSTH precedentemente fissati dovranno risultare, come da listato 2. Solo sotto queste condizioni il microcontrollore riconoscerà il bit "0" o il bit "1" inviato dal trasmettitore. Il listato 3, anche se non completo per ragioni di spazio, mostra ciò che è stato affermato. Ovviamente tale operazione si ripeterà per tutti gli impulsi provenienti dal trasmettitore, nell'ambito della trama, opportunamente inizializzata dall'interrrupt generato dal multivibratore monostabile 74HC123. La variabile chiamata impulsi, verrà incrementata ad ogni fronte di salita, dell'impulso in arrivo da parte del trasmettitore, e permetterà di associare i diversi bit ricevuti,ad esempio, ai minuti alle ore, eccetera. Un'altra caratteristica del DCF77, è che nell'ambito di qualsiasi parte della trama inviata, ad esempio le ore, il primo bit inviato è sempre quello meno significativo, il che comporta una ulteriore complicazione nella gestione delle informazioni ricevute.

Gestione di inizio ricezione

Per la gestione dell'inizio ricezione, ovvero per la sincronizzazione con il trasmettitore, utilizzeremo l'interruzione scatenata dal multivibratore in

ENJOY OUR MIX

La nostra ricetta: prendete un'azienda di grande tradizione con almeno 25 anni di successi all'attivo. Combinate insieme esperienza e passione. Aggiungete i migliori prodotti accuratamente selezionati per l'alimentazione di schede di controllo industriali e civili, o quelli per l'automazione e controllo. Terminate con la gamma di prodotti nel settore potenza e automazione. Ecco qui il grande mix di Elettromeccanica ECC.

Elettromeccanica ECC S.p.A.

Via F.lli Rosselli, 33
20090 Trezzano s/N (MI) Italy

Tel. +39 02 4844181
Fax +39 02 4450119

www.eccmec.it
vendite@eccmec.it

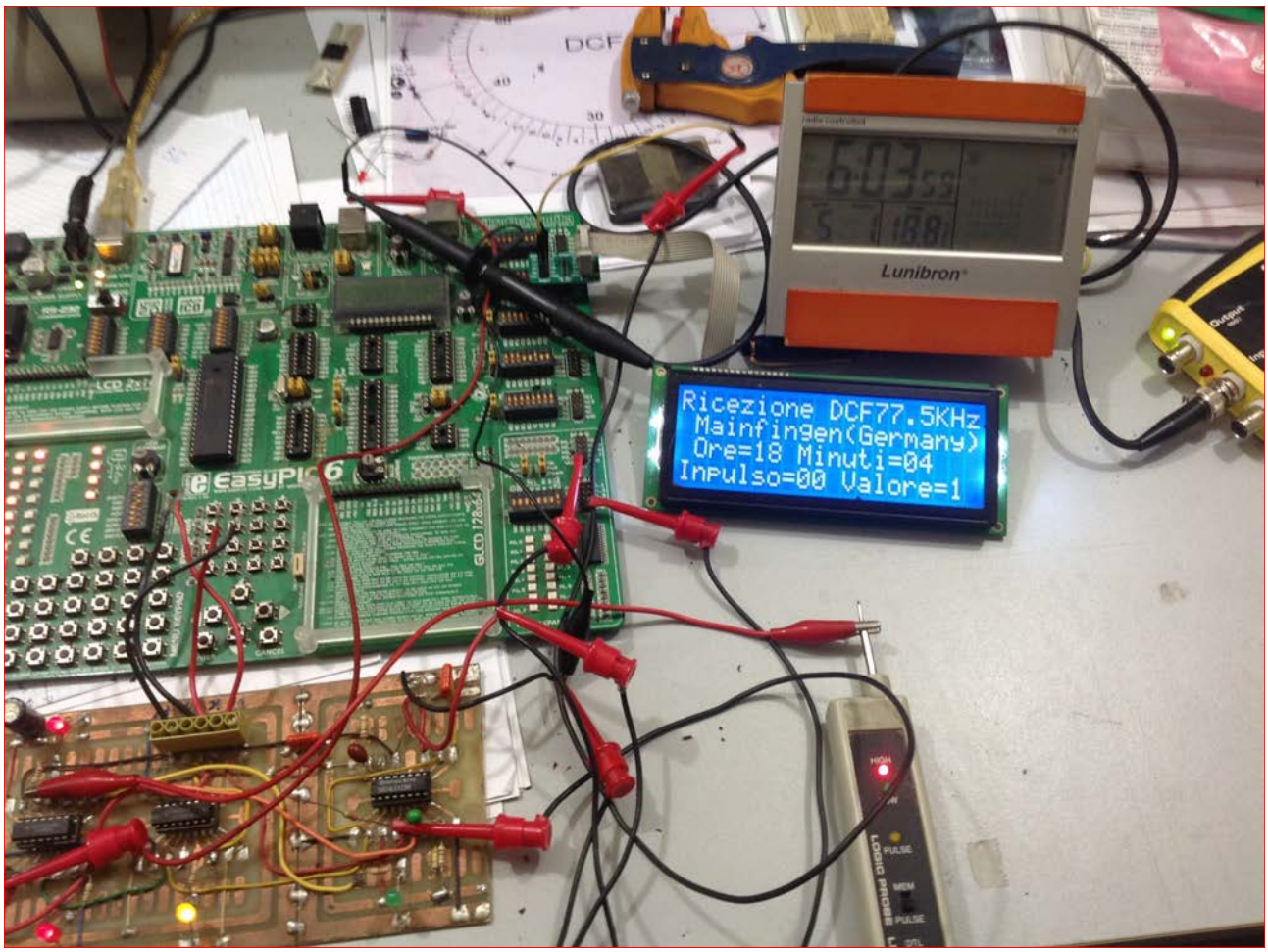


Figura 6: Realizzazione prototipo

corrispondenza del fine trama. L'interruzione è legata al fronte di salita che si presenta sul pin RB0, in questa fase andremo a controllare il valore della variabile "impulsi", se tale variabile contiene il valore "0" vuole dire che non si è inizializzato nessun conteggio precedente, ciò significa che l'inizializzazione della ricezione è corretta. Per ciò azzereremo tutte le variabili di lavoro e setteremo il flag di corretta inizializzazione insieme alla variabile syncro (syncro=0 non sincronizzato, syncro=1 sincronizzato) che verrà visualizzata sul display LCD. In caso contrario azzereremo ancora le variabili di lavoro (tra cui impulsi), insieme al flag di corretta inizializzazione,e al contempo setteremo il flag di errore trama, spengendo i tre led, di ricezione minuti, ricezione ore, ricezione datario. La gestione corretta dell'inizio ricezione è effettivamente molto importante, al fine di evitare errori casuali, che potrebbero presentarsi al momento in cui il segnale del trasmettitore DCF77, fosse fortemente disturbato per ragioni di tipo metereologico o da disturbi elettromagnetici presenti lungo la tratta di ricezione. Il listato 4 evidenzia ciò che testé è stato affermato. Le foto seguenti mostrano la realizzazione del prototipo e le successive fasi di debug del software di sistema.

Osservazioni Finali

Oltre la soluzione da me adottata si sarebbe potuto usare una soluzione prettamente hardware, di cui riporto lo schema di principio mediante l'uso del simulatore Proteus. In questa soluzione il discriminatore temporale è costituito dal dispositivo NE555 e dal FLIP-FLOP di tipo D(74HC74), sono presenti anche due multivibratori monostabili retriggerabili, uno svolge la

Figura 7: La centralina

funzione di fine trama mentre l'altro indica al micro controllore quando andare a leggere il dato decodificato (dato pronto). Come dicevo all'inizio dell'articolo la decodifica del DCF77 non è un'impresa proibitiva, ma richiede un certo impegno, e questo è legato soprattutto alla fragilità della trasmissione indipendentemente dal tipo di soluzione adottata. Ricordo che, minore è la distanza che sussiste tra punto di trasmissione quello di ricezione e migliore è la qualità del segnale ricevuto. Se si vuol ricevere un segnale certo, che trasmetta l'orario, si deve far ricorso al segnale GPS, e in modo particolare alla stringa NMEA0185 con intestazione $GPGGA.

Bibliografia e riferimenti web

http://www.ecat.it/site/index.php?module=site&method=home

http://www.ti.com/product/cd74hc123

http://www.microchip.com/wwwproducts/Devices.aspx?product=PIC16F887

http://www.mikroe.com/mikroc/pic/

http://www.mikroe.com/easypic/v6/

https://www.youtube.com/watch?v=4QLvl0b0nrg

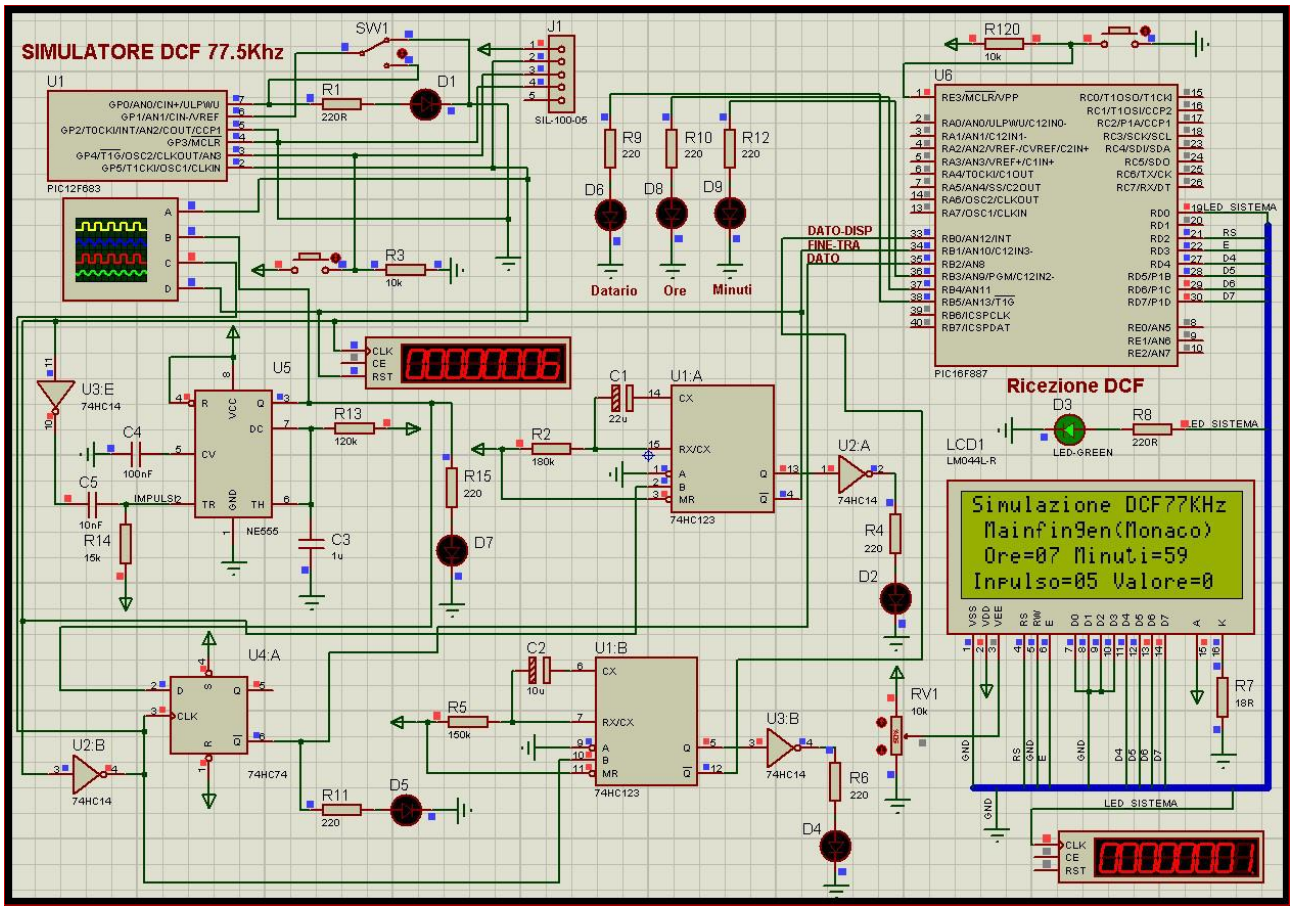


Figura 8: Soluzione hardware per la decodifica DCF77

SMART HUT ON THE ROAD

I ROADSHOW SMART ALLA SCOPERTA DELLA DOMOTICA

Gli eventi dedicati a domotica, sicurezza e tecnologie per la smart home

OBIETTIVI
- Incontro diretto con il proprio target
- Formazione e qualificazione professionale
- Confronto con gli esperti del settore

FORMAT
- *Conferenza plenaria:* coordinata da Smart Hut
- *Workshop aziendali* tecnici e di approfondimento
- *Area espositiva:* per gli incontri one-to-one

TARGET
- Installatori, impiantisti, system integrator
- Progettisti, architetti, geometri, periti
- Operatori e professionisti del settore

ARGOMENTI
- Perché la domotica è una opportunità per case e retail?
- Diminuire e gestire i consumi energetici: soluzioni e tecnologie
- Sicuri sempre, anche da lontano. Tutto sotto controllo in un clic
- Domotica senza fili e nuove applicazioni tecnologiche
- Orientarsi tra gli standard di comunicazione
- Come si fa luce a diventare intelligente?
- Integrazione di domotica, sicurezza e illuminazione nella progettazione di interni

TAPPE 2016
- > **TORINO**
- > **VERONA**
- > **FIRENZE**
- > **BERGAMO**
- > **BOLOGNA**
- > **TRENTO**
- > **ANCONA**
- > **ROMA**

> **PADOVA** 6-8 ottobre
ILLUMINOTRONICA

HUT assodel LUMEN CONSORZIO ELINT

I ♥ DOMOTICS

www.smarthut.it/roadshow

Generatore di frequenza con un PIC

di Giovanni Carrera

Argomento di questo progetto è un generatore di onda quadra la cui frequenza, impostata da un encoder, varia da 4 Hz a 20 kHz. Si propone un'applicazione con un microcontrollore PIC molto comune come il PIC16F628, collegato a un display LCD e a un encoder elettromeccanico con pulsante integrato.

Con un approccio diverso dal solito, anziché impostare il periodo dell'onda da generare, cosa più immediata a livello software, si è preferito impostare la frequenza da generare, perché più comoda per l'operatore. Questo crea una piccola difficoltà poiché occorre convertire la frequenza in periodo e, quindi, ricavare il numero intero da caricare nel timer del microcontrollore. La frequenza generata non sempre coincide con quella impostata, per via dell'arrotondamento da numero reale a intero. Per questo motivo, sul display sono visualizzate sia la frequenza richiesta sia quella generata. Con piccole variazioni sul programma è possibile modificare l'intervallo di frequenze generate, tenendo conto che la routine d'interrupt richiede qualche ciclo non trascurabile per le alte frequenze. Questo lavoro vuole essere anche un esempio applicativo dell'encoder incrementale usato come compatto ed economico dispositivo d'immissione di grandi numeri.

Generazione frequenza

Per la generazione di frequenza è utilizzato il Timer1, timer/counter a 16 bit, configurato in modo timer con clock interno. Il registro TMR1 è incrementato con il clock e a ogni overflow (superamento di $FFFF), se è abilitato l'interrupt sul Timer1, è settato l'interrupt flag bit PIR1.TMR1IF. Il periodo di interrupt da generare è impostato caricando la coppia di registri TMR1 (TMR1H e TMR1L) con un numero compreso tra 0 e 65535. La routine di interrupt da Timer1 cambia stato al bit RB3 d'uscita, quindi, questo periodo corrisponde al semiperiodo dell'onda quadra generata.

Il registro di controllo T1CON di Timer1 è impostato in questo modo:

•**TMR1CS** = 0 : ingresso dal clock interno, 1/4 di frequenza xtal = 4 MHz;
•**T1CKPS1** = 1, T1CKPS0 = 1 : Prescaler = 8;
•**TMR1ON** = 1: il Timer1 è abilitato.

Con queste impostazioni la frequenza d'ingresso di Timer1 è 4000/8 = 500 kHz, il che corrisponde a un periodo base di 2 µs, questa è la risoluzione temporale del sistema. Caricando il registro TMR1 con il numero NT1 = 65535, il periodo d'interrupt è di 0,002 ms ed essendo pari al semiperiodo dell'onda quadra generata, la sua frequenza massima sarà pari a 250 kHz, valore massimo teorico. Questo non tiene conto dei ritardi causati dalle istruzioni della routine d'interrupt e, soprattutto, dall'errore di arrotondamento che cresce con la frequenza, quindi è preferibile limitarla a 20 kHz. Il valore minimo di frequenza generata è di circa 3,8147 Hz, corrispondente a NT1 = 0. Data la frequenza fr richiesta, il processore deve calcolare il numero NT1 da caricare nel registro TMR1:

$$NT1 = 65536 - INT\ (Tr / (2*tb))$$

Dove Tr = 1/fr è il periodo richiesto dell'onda quadra generata e tb = 2 [µs] è il periodo base corrispondente alla frequenza in ingresso al Timer1, ossia 500kHz. Esprimendo Tr in microsecondi si ottiene: Tr = 1e6/fr , dividendo per due, essendo un semiperiodo, e per tb si ottiene l'espressione:

$$NT1 = 65536 - INT\ (250000 / fr)$$

F. richiesta [Hz]	F. Generata [Hz]	F. misurata [Hz]	T misurato [ms]	T generato [ms]	Deltat [ms]	Errore %
10	10	10	100.00000	100.00000	0.00000	0.00%
50	50	50	20.00000	20.00000	0.00000	0.00%
100	100	100	10.00000	10.00000	0.00000	0.00%
500	500	497	2.01207	2.00000	-12.07243	0.60%
1000	1000	988	1.01215	1.00000	-12.14575	1.20%
5000	5000	4705	0.21254	0.20000	-12.53985	5.90%
10000	10000	8887	0.11252	0.10000	-12.52391	11.13%
15000	15625	13069	0.07652	0.06400	-12.51695	16.36%
20000	20833	16525	0.06051	0.04800	-12.51360	20.68%

Tabella 1: Risultati delle misure temporali prima delle correzioni

F. richiesta [Hz]	F. Generata [Hz]	F. misurata [Hz]	T misurato [ms]	T generato [ms]	Deltat [ms]	Errore %
10	10	10	100.00000	100.00000	0.00000	0.00%
50	50	50	20.00000	20.00000	0.00000	0.00%
100	100	100	10.00000	10.00000	0.00000	0.00%
500	500	500	2.00000	2.00000	0.00000	0.00%
1000	1000	1000	1.00000	1.00000	0.00000	0.00%
5000	5000	4987	0.20052	0.20000	-0.52136	0.26%
10000	10000	9948	0.10052	0.10000	-0.52272	0.52%
15000	15625	15501	0.06451	0.06400	-0.51197	0.79%
20000	20833	20614	0.04851	0.04800	-0.50995	1.05%

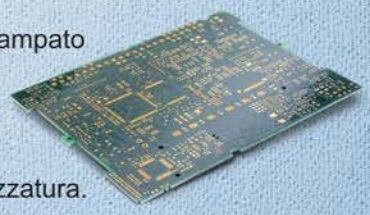
Tabella 2: Risultati delle misure temporali dopo le correzioni

Come già detto, l'arrotondamento necessario per ottenere il numero intero NT1, causerà degli errori, che cresceranno con l'aumentare della frequenza e si manifestano quando la divisione non dà un intero. Per questo, è ricalcolata la frequenza effettiva generata, con la formula inversa:

fg:= INT(250000/(65536-NT1))

Questa volta si potrebbero fare i calcoli in virgola mobile, ma non avendo spazio sul display e si è usato nuovamente un intero. Lo schermo LCD impiegato nel prototipo aveva una riga a 16 colonne, per cui la frequenza generata fg è mostrata insieme a quella richiesta fr, utilizzando 2 campi da 8 caratteri. Una volta realizzato il sistema e caricato il firmware, sono state eseguite le misure con un frequenzimetro e si è notato che il periodo generato era maggiore di quello ricavato dalla frequenza generata, indicata sul display da circa 500 Hz in su e che il ritardo era quasi costante e pari a circa 12,5 µs. Questo è dovuto ai cicli della CPU della routine di interrupt. A questo punto si può operare una correzione aumentando (perché è il complemento a 65536) di 3 unità il valore calcolato per

AVERE DEI CIRCUITI STAMPATI NON È MAI STATO COSÌ FACILE



Oggi con **pcboards** puoi ordinare **online** i tuoi prototipi di circuito stampato ed averli in due giorni, qualsiasi sia la sua complessità, al miglior prezzo sul mercato. È facile! Entra in **pcboards.eu**, scegli il servizio che desideri ricevere e inserisci le caratteristiche del tuo circuito. Otterrai un preventivo istantaneo senza costi di attrezzatura. **Pcboards** garantisce standard qualitativi massimi perché fa riferimento ad un sito produttivo tecnologicamente all'avanguardia, presente sul mercato da oltre vent'anni.

- **ECONOMICO**
- **CONSEGNATO IN DUE GIORNI**
- **DI QUALITÀ GARANTITA**

pcboards.eu
get your pcb online

www.pcboards.eu - info@pcboards.eu

narrativita

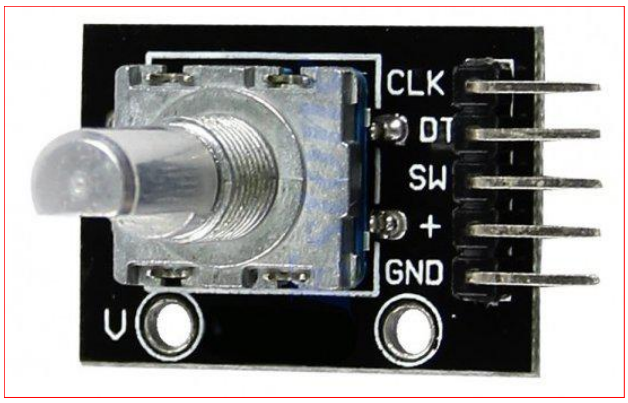


Figura 1: Aspetto tipico del modulo encoder.

NT1, ossia, riducendo il semiperiodo di 6 µs. La tabella 1 mostra i risultati delle misure fatte sul prototipo prima della correzione, dove l'errore è calcolato tra la frequenza generata e quella misurata e deltat= (T generato - T misurato)*1000. Questa tabella è interessante anche perché mostra alcuni casi in cui vi è differenza tra la frequenza richiesta e quella generata. Questa differenza esiste anche per frequenze inferiori a 10kHz, per tutti quei valori che danno un resto nella divisione. L'errore sulla frequenza è inaccettabile già a soli 1000 Hz, per cui si è rimediato con NT1= NT1+3. Dopo la correzione vista prima, sono state rifatte le misure e le cose sono notevolmente migliorate, come mostra la tabella 2. Come ci si poteva aspettare, il deltat è ora di circa 0,5 µs, purtroppo non ulteriormente correggibile, e gli errori sono accettabili.

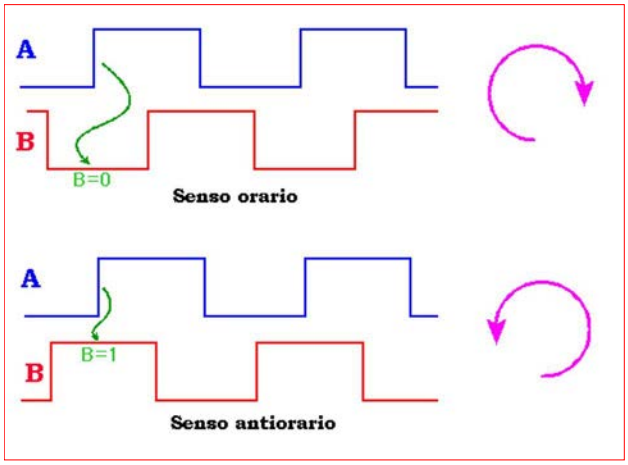


Figura 2: Segnali di un encoder incrementale.

Gli interrupt

Il PIC16F628 ha 10 sorgenti di interrupt, tre di esse sono utilizzate per questo progetto:

- Interrupt esterno RB0/INT, per l'encoder;
- Interrupt su variazione PORTB (4..7), RB4 per il pulsante dell'encoder;
- Interrupt su overflow di TMR1, per la generazione dell'onda quadra.

L'interrupt esterno sul pin RB0/INT è di tipo edge triggered: esso può essere generato da fronte di salita o di discesa del segnale, impostando il bit OPTION. INTEDG, rispettivamente a 1 o 0. Quando avviene la transizione impostata sul pin RB0/INT, è settato il flag interrupt bit INTCON.INTF, che sarà poi azzerato dalla routine di interrupt da RB0. L'interrupt su variazione PORTB è generato da un cambiamento di ingresso sui bit <7: 4> di PORTB e setta il suo flag interrupt bit INTCON.RBIF, che sarà poi azzerato dalla routine di interrupt

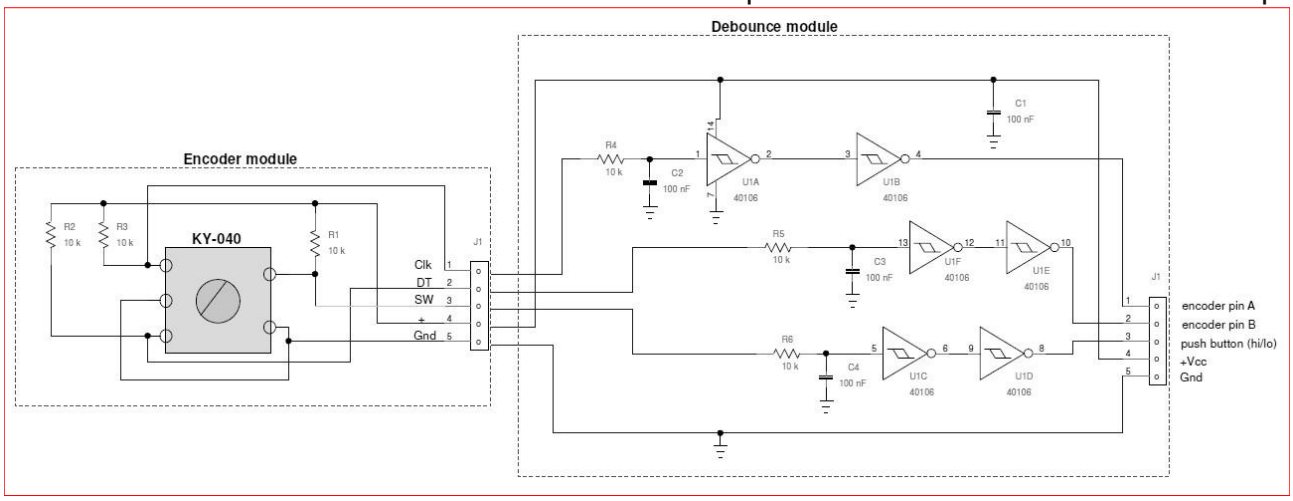


Figura 3: Schema dei circuiti anti-rimbalzo dell'encoder.

da variazione di RB4. L'interrupt può essere attivato/disattivato impostando il bit INTCON.RBIE. Solo i bit configurati come input possono provocare l'interrupt, per cui si è abilitato solo il bit PORTB .4.

L'encoder

Per l'immissione della frequenza da generare si utilizza un piccolo encoder elettromeccanico, come quello mostrato in figura 1. Esso è montato su una piccola scheda su cui sono saldati tre resistori smd di pull-up da 10 kOhm. Naturalmente è possibile acquistare il solo encoder e incorporarlo nella scheda anti-rimbalzo, ma nel prototipo si è usato questo modulo. L'encoder incrementale ha al suo interno un disco conduttore, solidale al perno. Su questo disco sono praticate delle fessure in modo da realizzare una specie di ruota dentata. Un paio di contatti striscianti, opportunamente disposti, permette di generare due segnali sfasati di 90° in modo da ricavare il senso di rotazione del perno, come mostrato in figura 2. L'encoder utilizzato è anche munito di un pulsante che scatta premendo il perno e che sarà utilizzato per posizionare la cifra. Per l'encoder sono necessari due interrupt, uno per la decodifica dei segnali e l'altro per il pulsante. Il canale A è collegato a RB0/INT, impostato su fronte di salita. La routine di interrupt leggerà il canale B e porrà delta=+1 se B=0 (senso orario: incremento) oppure delta=-1 se B=1 (senso antiorario: decremento). La variabile delta è usata anche come flag di interrupt da encoder, se delta=0, l'encoder non è stato mosso e tutto rimane invariato. Il secondo interrupt, quello su variazione di RB4, è usato per il pulsante dell'encoder, a ogni schiacciamento della manopola deve generarsi un interrupt che incrementerà l'indice della tavola con i moltiplicatori dell'incremento totale. Poiché a ogni

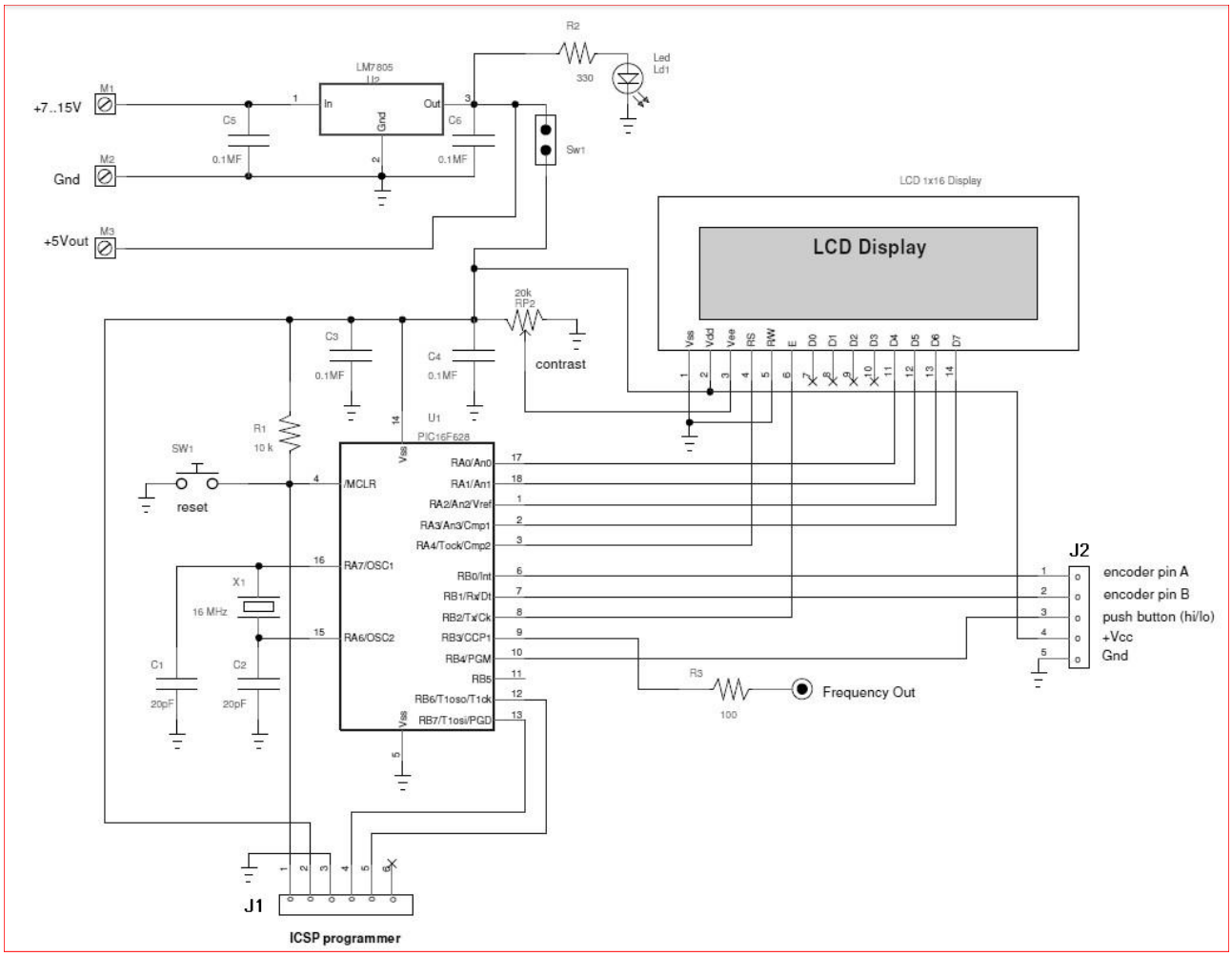


Figura 4: Schema della scheda madre.

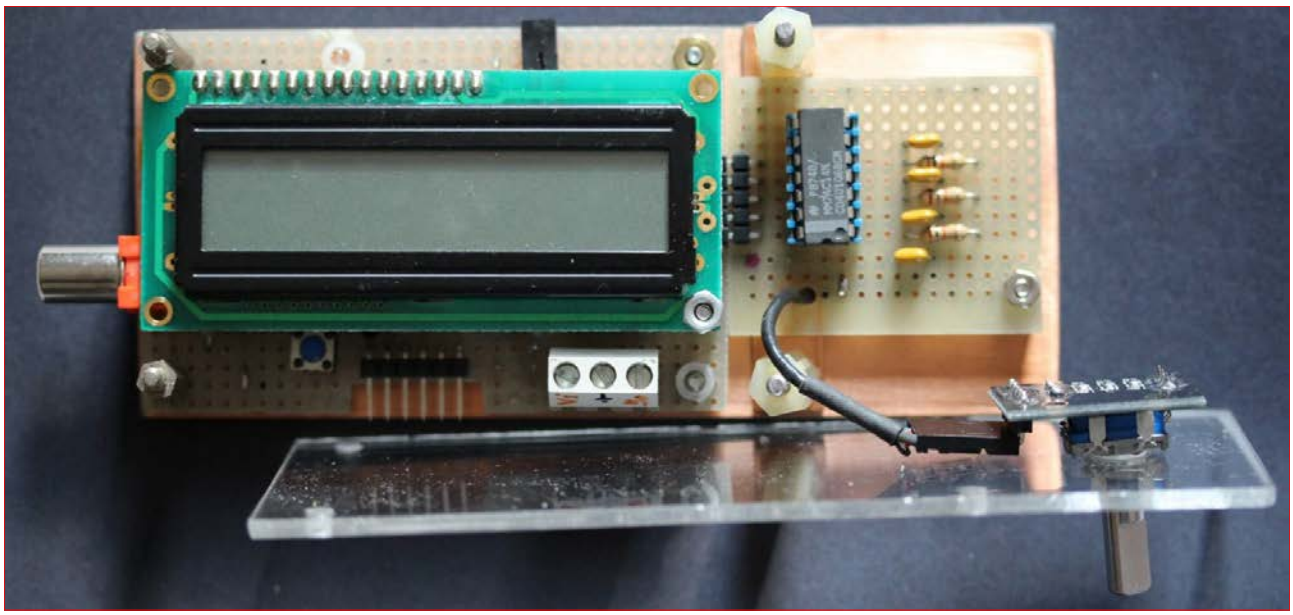


Figura 5: Il prototipo.

pressione della manopola ci sono due cambiamenti di RB4, per identificare la transizione da zero a uno, si fa una lettura di RB4. Se questa è uno si procede a incrementare l'indice del moltiplicatore, che varia da 1 a 5, ciclicamente. Premendo la manopola il moltiplicatore passa da uno a dieci, premendo nuovamente passa a 100, poi a 1000, poi a 10000, quindi ritorna a 1 con un ulteriore schiacciamento del pulsante. Al reset il moltiplicatore è uno e la frequenza richiesta è pari a 1000 Hz, valore di default. Se la frequenza immessa è superiore a 20kHz o inferiore a 4 Hz, è impostato il valore minimo, ma questa frequenza può essere cambiata come pure quella di default.

L'hardware

Per realizzare questo strumento è stato utilizzato un PIC16F628 interfacciato a un piccolo display LCD 1x16, che è gestito dal suo controller come un display da 2 righe da 8 caratteri. Questa scelta è giustificata dal fatto che la scheda era già stata realizzata per precedenti progetti ed è stata modificata per quest'applicazione. Con modeste variazioni al programma è meglio usare un display da 2x16, che permette di inserire maggiori informazioni e rappresentare numeri in virgola mobile per la frequenza effettivamente generata. Per l'encoder è stato realizzato un circuito anti-rimbalzo perché la sua gestione software è fatta con gli interrupt e deve essere più snella possibile, soprattutto in questo caso specifico. La figura 3 mostra la il circuito, montato nel prototipo su una scheda separata, ma che può essere incorporato nella scheda del PIC. Lo schema della scheda principale è mostrato in figura 4. La scheda ha un alimentatore a 5V realizzato con un comunissimo 7805, che potrebbe anche essere un 78L05, visto che il sistema consuma circa 5 mA. Il connettore J1 serve per programmare il PIC con un programmatore PICKIT2 o simili. Il connettore J2 serve per interfacciare la scheda anti-rimbalzo dell'encoder. Il ponticello Sw1 va tolto in programmazione e inserito nel normale funzionamento. La foto di figura 5 mostra il prototipo con il pannello frontale trasparente smontato. La scheda dell'encoder è fissata al pannello

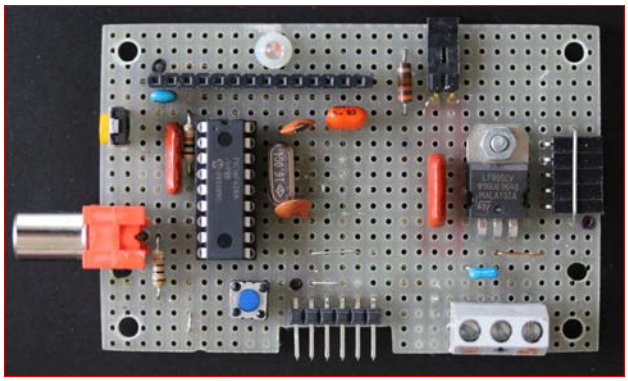
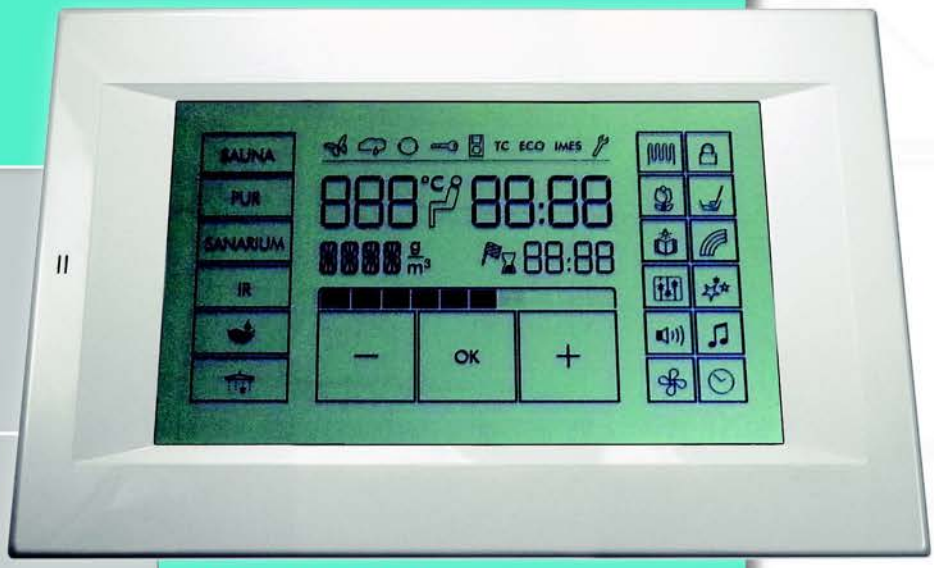


Figura 6: Disposizione dei componenti.

LCM CHAR
LCM GRAPH
OLED
TFT MONITOR
TFT PANEL
VFD

DISPLAY

Applicazioni HOME, MEDICALE e PANNELLI DI CONTROLLO



EMBEDDED &IPC

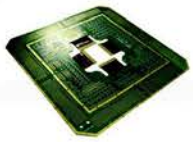
OPTOELETTRONICA

INCELL
Touch Display

LCD MONOCROMATICO CON FUNZIONI TOUCH INTEGRATE

Questa nuova tecnologia sostituisce il tradizionale LCD più l'addizionale touch screen e relativo controller.

- RISPARMIO SUI COSTI DEI MATERIALI
- SPESSORE RIDOTTO
- DUREZZA SUPERFICIE 3H
- NUMERO MASSIMO TASTI 20~24
- DIMENSIONE MINIMA ICONA/TASTO 8X8mm
- Top -20~70 °C
- PURE ICON 1/4 DUTY MAX
- DOT MATRIX+ICON 1/96 DUTY MAX



PRINTED CIRCUIT BOARD

DISPLAY CUSTOM

Mc'TRONIC

Il riferimento sicuro per l'innovazione tecnologica

Display & Embedded Computing Solutions

Operativa nel settore industriale della visualizzazione (**Display LCD**), dei **Sistemi Embedded** e **SBC**.
Il **know how**, fondamentale in un ambiente altamente tecnologico ed in **costante evoluzione**, è stato assiduamente coltivato per **oltre 25 anni**, investendo nelle **persone** e nella **ricerca**.

Mc'Tronic S.r.l.
Sede amministrativa - Corso Milano, 180 - 28883 - GRAVELLONA TOCE (VB)
T. +39 0323 86931 r.a. - F. +39 0323 869322
Sede legale - Via Novara, 35 28010 VAPRIO D'AGOGNA (NO)
www.mctronic.it - info@ mctronic.it

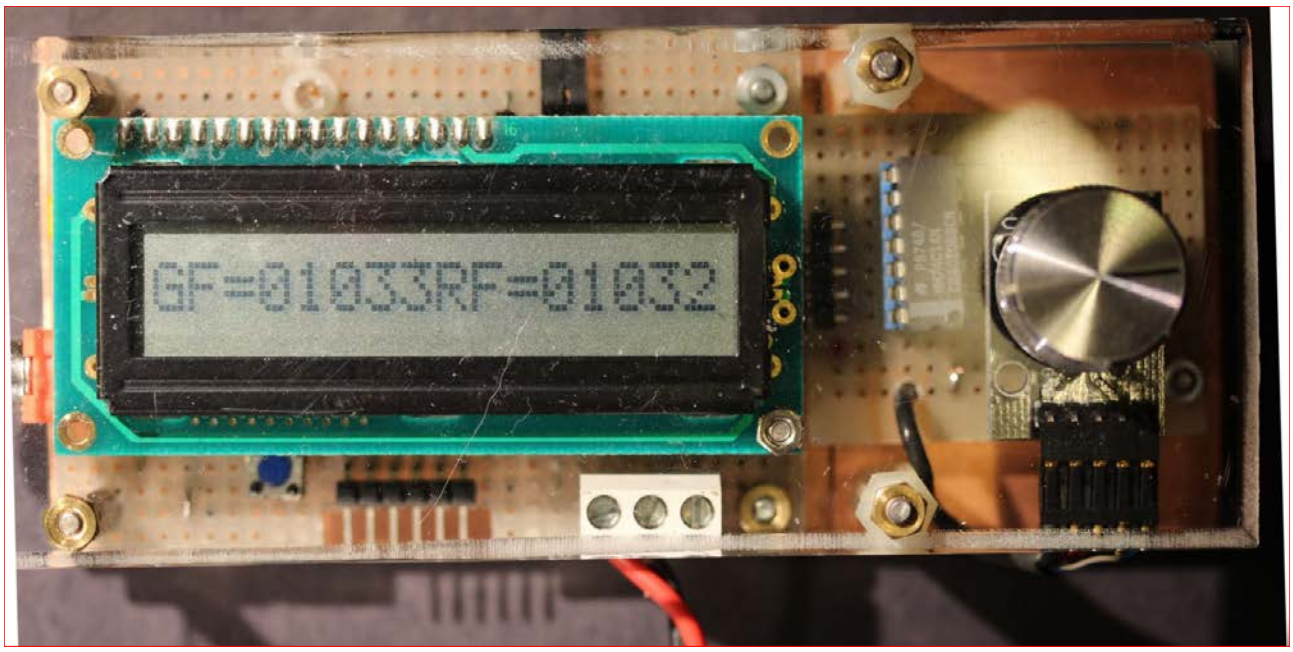


Figura 7: Prototipo montato e in funzione.

con la vite filettata della boccola dell'encoder e collegata con un piccolo connettore. E la figura 6 mostra la sola scheda madre, con una millefori un poco più grande o con una disposizione più compatta si potrebbero montare anche i componenti dei circuiti anti-rimbalzo. A sinistra si nota il connettore RCA per l'uscita dell'onda quadra e il trimmer per la regolazione del contrasto del display. In basso il connettore per la programmazione ICSP del PIC. La figura 7 mostra il sistema in funzione e completamente montato. Si possono chiaramente notare le due frequenze: a sinistra quella effettivamente generata (GF) e a destra quella richiesta (RF). In questo caso le frequenze differiscono di una sola unità.

Il programma

Il programma è stato scritto in Pascal, usando un potente cross-compiler su PC Windows, il mikroPascal PRO per PIC. Questo compilatore è distribuito in Italia da Elettroshop come pure altri prodotti di mikroElektronika. Presso il sito www.mikroe.com è possibile scaricare gratuitamente una versione completa, ma limitata a soli 2k word di codice prodotto. In questo caso il codice ha una lunghezza inferiore, quindi, si può compilare senza problemi. Chi ha più dimestichezza con il C o il Basic, può scaricare questi compilatori e tradurre il programma nel linguaggio desiderato. Se questo programma fosse stato scritto per Arduino, l'occupazione di memoria flash sarebbe stata molto maggiore, come pure il tempo di compilazione. Questo è dovuto ai linguaggi usati che, pur rendendo più facile la vita al programmatore del compilatore, producono un codice non ottimizzato a livello di codice macchina. Il programma è sufficientemente commentato (in inglese) ed è già stato descritto nei precedenti paragrafi. Dato il limitato spazio sul display, le due scritte sono attaccate e si è dovuto sovrascrivere il primo zero con '=' in quanto la stringa della conversione da intero con la funzione IntToStrWithZeros(gfreq, texd) è sempre di 6 caratteri, indipendentemente dal numero. Con un display più grande le due frequenze sarebbero su due righe separate e ci sarebbe più spazio per le scritte. Con il PIC16F628 e simili occorre disabilitare il comparatore analogico per permettere di usare i pin da esso impiegati come bit di I/O digitali.

Si noti che il numero 65536 è stato scritto come 65535+1 perché non rappresentabile con variabile intera senza segno (type: word). Per rendere più veloce l'esecuzione è meglio, nelle formule, sostituire i numeri con i nomi di costanti.

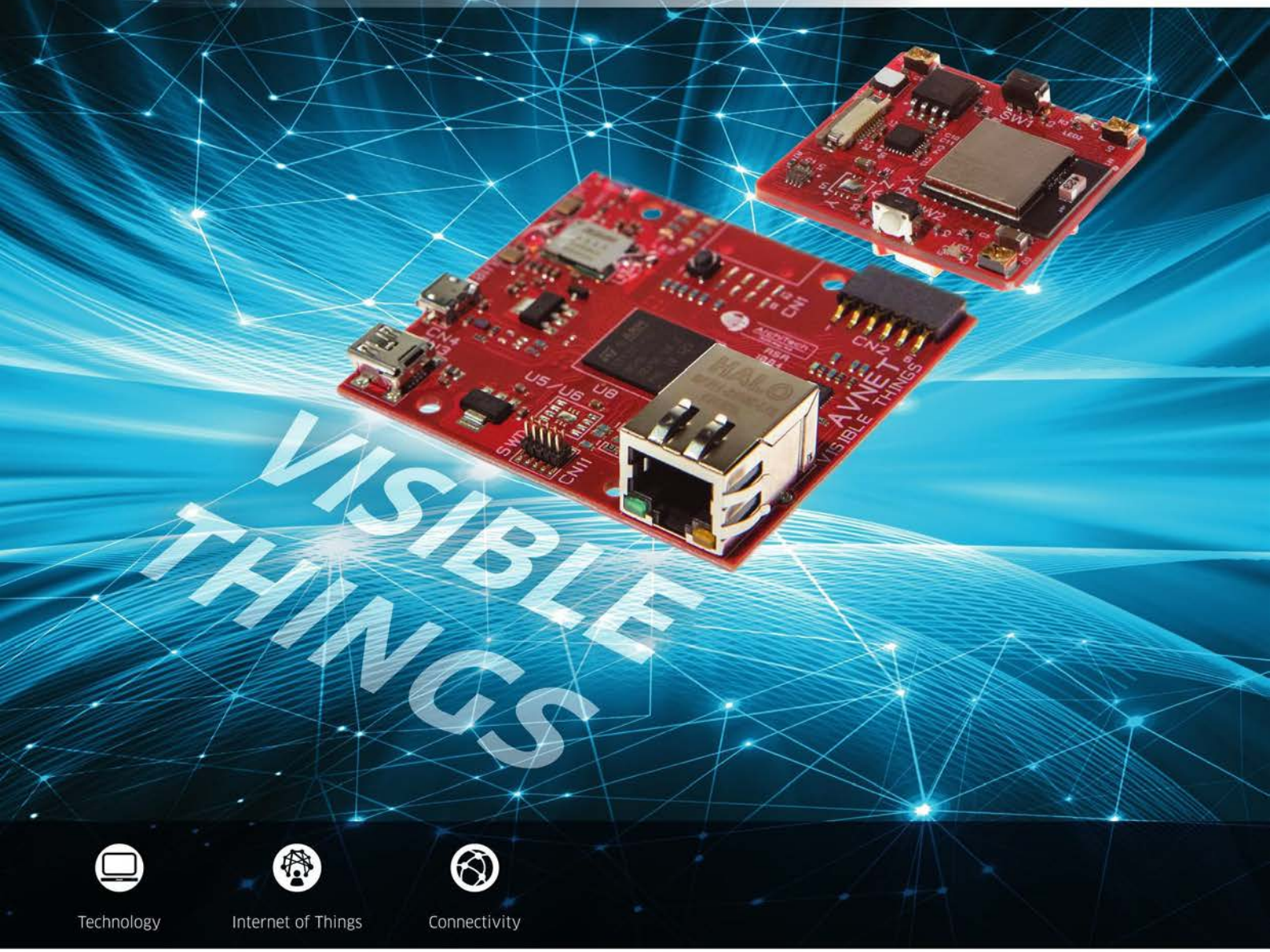
Listato

```
program FreqGenerator;
// Frequency generator with a PIC16F628 interfaced to an incremental encoder
// print on LCD display the requested frequency
// and the effective generated frequency
// timer#1 for base frequency interrupt
// I/O configuration:
// PortB.0= encoder A with interrupt on falling edge   (in)
// PortB.1= encoder B   (in)
// PortB.2= LCD E (out)
// PortB.3= Frequency output (out)
// PortB.4= Encoder push button (in)
// PortA.4= LCD RS (out)
// PortA.0..3= LCD data (4 bit mode) (out)
// xtal = 16.00 MHz
// the LCD display has 1x16 but is seen as 2 rows x 8
// by Giovanni Carrera 09/11/2015


var
  delta, multi,rfreq,gfreq: integer;
  NT1: word;
  texd: string[6];
  const multiplier : array[1..5] of integer = (1,10,100,1000,10000);
  const fremax : longint = 250000;
  const wmax : word = $FFFF;

// Lcd module connections
var LCD_RS : sbit at RA4_bit;
var LCD_EN : sbit at RB2_bit;
var LCD_D4 : sbit at RA0_bit;
var LCD_D5 : sbit at RA1_bit;
var LCD_D6 : sbit at RA2_bit;
var LCD_D7 : sbit at RA3_bit;
var LCD_RS_Direction : sbit at TRISA4_bit;
var LCD_EN_Direction : sbit at TRISB2_bit;
var LCD_D4_Direction : sbit at TRISA0_bit;
var LCD_D5_Direction : sbit at TRISA1_bit;
var LCD_D6_Direction : sbit at TRISA2_bit;
var LCD_D7_Direction : sbit at TRISA3_bit;
// End Lcd module connections


procedure interrupt;
// interrupts on Timer#0 and on falling edge of RB0
begin
  if TestBit(PIR1,TMR1IF) then  // timer#1 interrupt
    begin
      TMR1L := Lo(NT1); // load the timer Lo
      TMR1H := Hi(NT1); // load the timer Hi
      PORTB.3:= not PORTB.3; // invert output
      PIR1.TMR1IF := 0; // clear Timer1 interrupt flag
    end
  else if TestBit(INTCON,INTF) then // interrupt on RB0
    begin
      // increment or decrement counter on PB1 status
      if TestBit(PORTB,1) then delta:= -1 else delta:= +1;
      PORTB.3:= not PORTB.3; // toggle led
      INTCON.INTF:= 0;// clear RB0 interrupt flag
    end
  else if TestBit(INTCON,RBIF) then // interrupt on RB4 Change
    begin
      if (PORTB.4 = 1) then
```

```
        begin // increment every 0 to 1 of RB4 change
          multi:= multi+1;// button interrupt, increment multiplier index
          if multi=6 then multi:=1; // back to top
          INTCON.RBIF:= 0;// clear interrupt RB4 change flag
        end;
    end;
end;

procedure IOinit;
// initialize I/O and variables

begin
  TRISB:= %00010011;  // PORTB bits 2 and 3 are outputs
  TRISA:= %10100000;  // PORTA bits 0..4 are outputs
  CMCON:= $07; // comparators off (RA0..RA4 usable as digital IO)
   multi:= 1;
  Lcd_Init(); // Initialize LCD on PORT A,B
  Lcd_Cmd(_LCD_CURSOR_OFF); // Turn off cursor
  ClearBit(OPTION_REG,INTEDG);// Interrupt on falling edge of RB0
  SetBit(INTCON, RBIE); // Enables the RB port change interrupt
  NT1:= $FF06; // load the timer  (for a 1000 Hz square wave)
  TMR1L := Lo(NT1); // load the timer Lo
  TMR1H := Hi(NT1); // load the timer Hi now
  T1CON:= $31; // Timer1 enabled and Prescaler = 8
  SetBit(T1CON,TMR1ON); // Enables Timer1
  PIR1.TMR1IF := 0;// clear Timer1 interrupt flag
  INTCON.INTF:= 0;// clear RB0 interrupt flag
  INTCON.RBIF:= 0;// clear interrupt RB4 change flag
  SetBit(INTCON, INTE); // Enables RB0 interrupt
  SetBit(PIE1, TMR1IE);// Enables Timer1 interrupt
  SetBit(INTCON, PEIE); // Enables peripheral interrupt
  SetBit(INTCON, GIE); // Enables global interrupt
  rfreq:= 999; //start value - 1
  Lcd_Out(1, 1, 'FreqGen'); // Print text to LCD (colums 1..8)
  Lcd_Out(2, 1, 'by GCar'); // Print text to LCD (colums 8..16)
  Delay_ms(2000);
  Lcd_Cmd(_LCD_CLEAR); // lcd clear
end;

begin { Main program }
  IOinit; // Initialize
  Lcd_Out(1, 1, 'GF');// generated frequency
  Lcd_Out(2, 1, 'RF');// requested frequency
  delta:= 1; // the first time is set by the program
   while true Do //  endless loop
    begin
      if delta<>0 then // encoder is been turned
        begin
          rfreq:= rfreq + delta*multiplier[multi];// increment or decrement frequency
          if (rfreq > 20000) Or (rfreq < 4) then rfreq:=4;// range control
          delta:= 0;// resets the flag
          NT1:= wmax - integer(fremax/rfreq)+1;// evaluate TMR1
          IntToStrWithZeros(rfreq, texd); // convert to string
          texd[0]:= '=';
          Lcd_Out(2, 3, texd); // Print text to LCD (colums 8..16)
          gfreq:= integer(fremax/(wmax-NT1+1)); //evaluate generated frequency
          NT1:= NT1+3;// this correction to take account of the delay of the program
          IntToStrWithZeros(gfreq, texd); // convert to string
          texd[0]:= '=';
          Lcd_Out(1, 3, texd); // Print text to LCD (colums 8..16)
        end
   end;
end.
```

AVNET Memec

SILICA An Avnet Company



Technology Internet of Things Connectivity

La galassia dell'IoT ha una nuova stella

Tieniti forte. Sta per decollare il tuo prossimo progetto edge-to-enterprise.

Visible Things™ è una piattaforma di valutazione e sviluppo altamente flessibile per progetti edge-to-enterprise nell'ambito dell'Internet delle cose (IoT). Il nostro team di esperti IoT ha creato Visible Things con un singolo obiettivo in mente: offrire uno strumento basato sulla tecnologia più avanzata per agevolare lo sviluppo di progetti IoT. Senza soluzione di continuità – dal sensore al server.

Rivolta a un'ampia gamma di mercati industriali, questa piattaforma include hardware testato, comprovato, sicuro e integrato e software incorporato per consentire di collegare sensori intelligenti e dispositivi embedded tramite soluzioni gateway o tecnologie di rete LPWAN (low power wide area network) direttamente alle applicazioni software in ambiente cloud ed enterprise. Questa piattaforma è offerta in esclusiva da Avnet Memec – Silica. Visita www.avnetmemec-silica.com/visible-things per scoprire tutti i dettagli sulla piattaforma e sui starter kit.

Starter Kit
(Gateway & Smart Sensor Board)
disponibile ora
199 €

Benvenuti nell'universo dell'IoT! #visiblethings

www.avnetmemec-silica.com | visible-things@avnet.eu

Accelerating Your Success™

Orologio-sveglia con Arduino

di Girolamo D'Orio

Realizzazione di un orologio sveglia che non teme i Blackout momentanei. Mantiene in memoria l'orario corrente, di allarme sveglia e lo stato di attivazione sveglia. Sono previste tre realizzazioni: un modulo RTC con visualizzazione orario su 4 display 7 segmenti, un modulo per ricezione segnale DCF77 e visualizzazione ora e data su LCD, e un modulo RTC e visualizzazione orario su 4 Nixie.

Comprare un comune orologio-sveglia sono capaci tutti. L'obbiettivo prefissato è quello di costruire una sveglia che risponda ai seguenti requisiti:

1-Precisione nella misura del tempo;
2-Mantenga orario e impostazioni dell'orario allarme in caso di Blackout.

Propongo in questo articolo tre realizzazioni basate sul famoso microcontrollore ATmega328p-pu. La prima realizzazione è basata su l'uso di un RTC (Real Time Clock) che ha il compito di contare il tempo anche quando viene a mancare l'energia elettrica. L'ora viene visualizzata su display a 7 segmenti. La seconda realizzazione sfrutta un modulo per la ricezione del segnale DCF77. Il segnale trasmette l'orario assoluto scandito da tre orologi atomici. La visualizzazione di ora e data avviene tramite Lcd. La terza realizzazione, si differenzia dalla prima solo nella visualizzazione dell'orario. Questa volta l'ora corretta sarà visualizzata dalle NIXIE. Parliamo della prima realizzazione, il modulo RTC ha il compito di tenere l'orario corretto e di inviarlo tramite comunicazione I2C al microcontrollore. Il microcontrollore ha il compito di ricevere l'orario e di pilotare i 4 display in modalità Multiplexing. Nel circuito sono presenti 4 pulsanti. P1 e P2

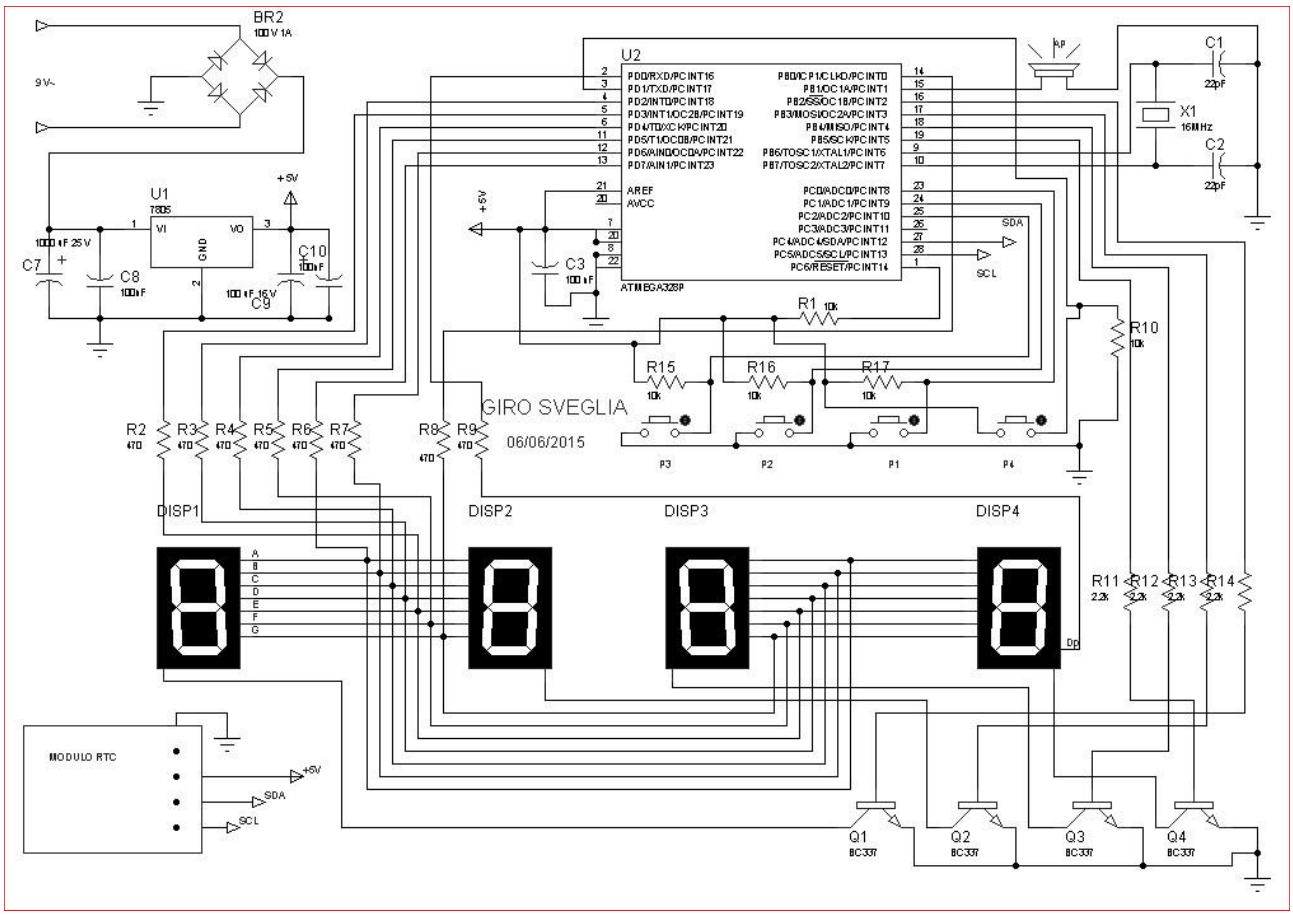


Figura 1: Schema orologio–sveglia con display 7 segmenti

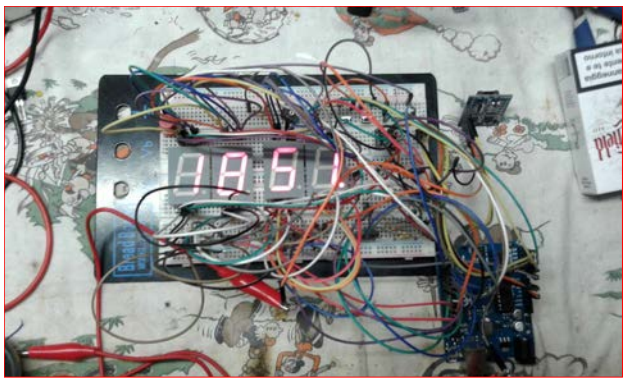


Figura 2: Montaggio provvisorio

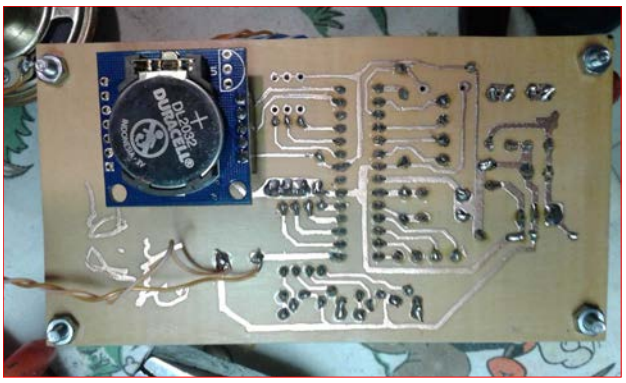


Figura 4: Vista dal retro

incrementano rispettivamente ore e minuti quando si entra in modalità settaggio orario sveglia. P3, se premuto entra nel settaggio dell'orario della sveglia. Una volto settato l'orario desiderato, per salvare l'ora della sveglia nella EEprom occorre premerlo di nuovo. P4, se premuto abilita/disabilita l'allarme sveglia impostato. Il punto decimale posto nel quarto display indica se l'allarme è attivato/disattivato. Inoltre ha la funzione di spegnere la suoneria quando suona. L'altoparlante dell'allarme l'ho recuperato da un vecchio giocattolo di mia figlia, ha come caratteristiche 100 Ω 0.1W. Mi sono limitato solamente ad inviargli un segnale ad onda quadra per fargli emettere un suono molto simile alle comuni sveglie. Guardando lo schema con i suoi numerosi collegamenti, dato che sfrutto 27 dei 28 pin del microntrollore, può sembrare un progetto complicato che comporterà un grosso ingombro nella realizzazione del Pcb. Mentre lo testavo su Breadboard facendo quindi molti collegamenti mi è venuto il dubbio anche a me di realizzare una sveglia di enormi dimensioni. La figura 2 mostra la realizzazione provvisoria su Breadboard. Sembra un gran groviglio (anzi, lo è...), i collegamenti sono numerosi e realizzare un Pcb monofaccia con il minor numero di ponticelli sembra difficile….. Mi è venuta l'idea di realizzarne 2, uno che ospita i display e l'altro il controllo. Per raggiungere l'obbiettivo di ottenere un oggetto il più piccolo possibile, i Pcb devono essere delle solite dimensioni, in modo di poterli montare uno dietro l'altro. Il risultato è accettabile, poco più grande di un pacchetto di sigarette, oltretutto ho posizionato il modulo RTC sul retro in modo da poterlo estrarlo facilmente, per esempio per sostituire la batteria tampone.

Descrizione del circuito Hardware

Il circuito ha un assorbimento modesto, qualsiasi piccolo trasformatore che rimediate può andare certamente bene, l'importante che il secondario abbia una tensione di uscita compresa tra 8-15V. Nel caso usate un trasformatore con un secondario superiore alle 12V consiglio di installare una piccola aletta di dissipamento al regolatore di tensione 7805. Non occorre necessariamente alimentare il circuito con un piccolo alimentatore stabilizzato, nel circuito è già presente. Con il microcontrollore non posso pilotare singolarmente tutti i led di tutti i

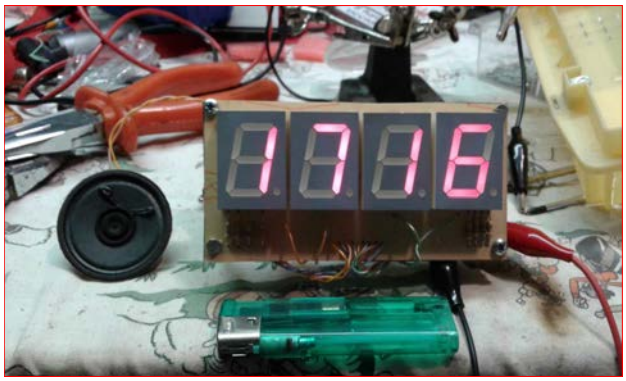


Figura 3: Ottimizzazione Display

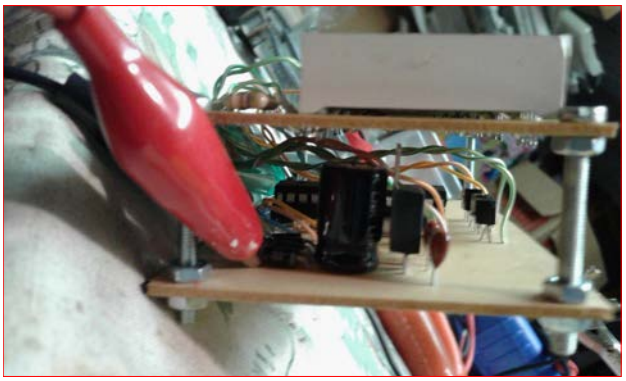


Figura 5: Vista frontale

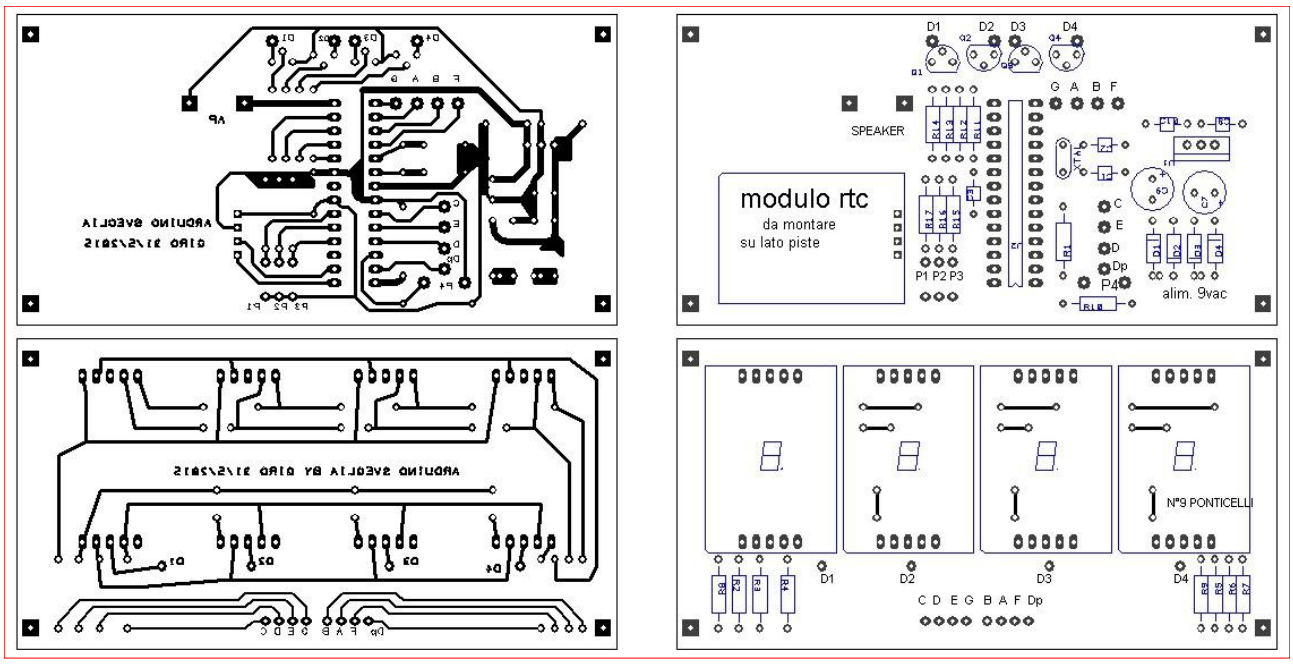


Figura 6: PCB e montaggio componenti

display, necessiterei almeno di 32 porte I/O e le relative resistenze per i led, quindi sfrutto la tecnica del Multiplexing. Questa tecnica ci permette di collegare in parallelo gli anodi dei display con solo 7 resistenze. Utilizzo 4 transistor NPN per pilotare i catodi di ciascun display. Facendo un rapido calcolo sfruttiamo così solo 11 porte I/O, quasi un terzo. Potremo però pilotare solo un display alla volta, quindi dobbiamo ingannare i nostri occhi per veder tutti i display accesi. Noto che il tempo di risposta dell'occhio umano è di circa 20 millisecondi tra un immagine e un'altra, se pilotiamo i display ad una velocità superiore, i nostri occhi percepiscono un immagine stabile e pulita. 20 millisecondi sono 1/50 di secondo, quindi 200 Hz. il microcontrollore che lavora con una frequenza data dall'oscillatore esterno con un quarzo di 16 Mhz, abbiamo un una velocità 80 volte superiore rispetto a questo limite, quindi non è un problema, vedremo dopo nel software la scelta adeguata.

Descrizione del software per Arduino

Il sorgente è ben commentato e non troverete nessuna difficoltà nel capirlo. Per prima cosa chiamo le librerie che ho fatto uso, in particolare la RTClib che con semplici righe ci permette di ricavare dal modulo omonimo ora e data. Per praticità ho inserito delle costanti per pilotare gli anodi dei display, ovvero ad ogni numero calcolato il microcontrollore sa già come portare i livelli logici I/O in relazione ai collegamenti con gli anodi (listato 1). La sub-routine chiamata display7 quando gli viene associata una variabile tramite la funzione switch porterà gli stati logici delle porte adeguati al numero da visualizzare, esempio del numero 3 (listato 2). Il microcontrollore adesso è pronto alla visualizzazione, ma c'è un problema, le variabili delle ore, minuti e secondi ottenute con la libreria sono un numero composto da

```
Elenco componenti

R2-R9          470 Ω ¼ W
R1,R15-R17     10 KΩ ¼ W
R11-R14        2,2 KΩ ¼ W
C7             1000 uF 24V elett.
C9             100 uF 16V elett.
C1,C2          22 pF poliestere
C8,C10         100 nF poliestere
U1             7805
U2             ATmega328p-pu
D1-D4          1N4007
Q1-Q4          BC337 NPN
XTAL           16 Mhz
Display 7seg   SC10-21SRW cat.com
TR1            Trasf. 0-9v 300mA
```

12 EDIZIONI DI SUCCESSO

500 OPERATORI DEL SETTORE

250 AZIENDE RAPPRESENTATE

42 INTERVENTI & WORKSHOP

POWER FORTRONIC

Il punto di incontro italiano per chi opera nel mondo dell'elettronica di potenza

Un'occasione unica di confronto tra produttori, fornitori e clienti sulle novità tecnologiche e sulle soluzioni innovative proposte sul mercato

Trend
Evoluzioni tecnologiche del mercato

Design
Analisi di progetto di sistema di potenza

Controllo motori
Pilotaggio, gestione e functional safety

Mobilità e Transportation
Soluzioni e tecnologie per trasmissione e conversione

Tecnologie
Componenti, strumenti di sviluppo e high power

Energy Storage
Energy harvesting, batterie ricaricabili e soluzioni

Thermal Management
Dispositivi per la dissipazione del calore

Efficienza Energetica
Moduli, soluzioni, architetture e topologie

TARGET

L'evento si rivolge a tecnici, progettisti, buyer e operatori del settore

PROMOSSO DA:

Associazione Italiana Distretti Elettronica

assodel

IN PARTNERSHIP CON:

Consorzio Elettrimpex
Lumen International

LUMEN CONSORZIO ELIMT

CONTATTI:

Tel. +39 02 210.111.236
marketing@tecnoimprese.it

TECNO IMPRESE

USA IL QR CODE
per registrarti e scoprire maggiori informazioni

partecipazione gratuita previa registrazione

http://fortronic.net/power

due cifre. Per visualizzare l'orario dato che i display sono quattro devo ricavare ogni singolo numero da inviare ad ogni singolo display. Con questo semplice algoritmo scompongo i numeri in modo da avere i numeri corrispondenti per le decine e le unità, sia delle ore e dei minuti (listato 3). Adesso siamo pronti per visualizzare l'orario con la tecnica del Multiplexing (listato 4). Come vedete è reso molto facile da capire, la sub-routine display7 associata alla variabile del numero da visualizzare, porta lo stato delle porte logiche adeguato al numero e il display relativo. La pausa di 3 ms. di accensione/spegnimento tra un display e l'altro permette di ingannare la nostra vista facendoci credere che tutti e quattro i display sono accesi, anche se in realtà non lo è. Se per curiosità volete capire come avviene il Multiplexing vi basta aumentare il valore inserito nel delay, rallentando riuscite a vedere come avviene. Passiamo adesso all'impostazione della sveglia. Premendo il tasto P3 viene richiamata un'altra sub-routine chiamata "setsveglia". Dato che voglio mantenere in memoria l'orario della sveglia impostato. Dopo averlo scelto l'orario di sveglia premendo i due pulsanti che incrementano le ore e i minuti, andremo a premere di nuovo P3 che registrerà in due celle della EEPROM l'orario selezionato. Eseguita questa operazione ritorna la visualizzazione dell'orario corrente (listato 5). Adesso premendo il pulsante P4 vedrete che il punto decimale del quarto display che rappresenta le unità dei minuti si accenderà. Conferma il corretto inserimento della sveglia. Per disabilitare la sveglia occorre semplicemente premere di nuovo P3 fino a che il punto decimale si spegne. Altra funzione di questo pulsante è quella di spegnere la suoneria della sveglia quando questa suona. Arrivato a questo punto mi sono posto un problema in caso di blackout momentaneo:

l'ora corrente è mantenuta dal modulo con la batteria tampone, l'impostazione dell'orario di sveglia è salvato nella EEprom, ma l'attivazione dell'allarme? Se accade un blackout momentaneo mentre sto dormendo il micro riavviandosi perderebbe tale impostazione, allora ho deciso di memorizzare nella eeprom lo stato dell'attivazione dell'allarme. Quando il micro verrà riavviato, nel void setup (), subroutine che viene eseguita solamente una volta, vado a leggere le celle della EEprom dove precedentemente era stato registrato l'orario della sveglia e lo stato dell'attivazione d'allarme. In questo modo sono sicuro che la sveglia suonerà.

Consigli per il montaggio

Per prima cosa vi consiglio di saldare i ponticelli dato che i grandi display verranno posizionati sopra. Nel Pcb sono presenti lettere adiacenti ai fori. I collegamenti con filo devono essere fati rispettando le lettere (Es. A-A, Dp-Dp, D1-D1). Essendo un produttore di vino, per deformazione professionale, ho "confezionato" il circuito su un contenitore in legno per bottiglie da 375 ml. I pulsanti sono posizionati sopra. Al centro in rilievo l'attivazione/disattivazione allarme, a desta il pulsante per incrementare i minuti e a sinistra il pulsante per incrementare le ore. Il pulsante per entrare nella modalità settaggio sveglia e posto dietro il pulsante centrale.

Il modulo RTC e la sua imprecisione

Montato il tutto a tarda sera e portato in casa la mattina mi sveglio con la mia Ardusveglia, confronto l'orario e vado a lavoro soddisfatto. Quando torno la sera confronto l'orario e noto che è avanti. Rimango un po' perplesso dato che il modulo doveva essere preciso. L'errore non poteva venire altro che dal modulo, ho dato la colpa immediatamente al quarzo da 32.768 Khz. Sostituito, recuperandolo da una vecchia sveglia ottengo quasi il solito errore nelle 24 ore, RTC si porta in avanti di quasi 30 secondi al giorno. Un errore inaccettabile. Collego il modulo al mio oscilloscopio dotato di frequenzimetro e qui la sorpresa, il quarzo oscilla ad una frequenza diversa (figura 8). La differenza, a patto che il frequenzimetro sia preciso,(data l'età) si discosta di 11 hz rispetto a quella teoricamente corretta. 11 Hz sono pochi, è solamente un errore dello 0.033%. Ma facendo due calcoli, dato che in un giorno

Listato 1

```
//anodo A collegato al pin 7
const int a=7;
const int b=6;
const int c=4;
const int d=2;
const int e=3;
const int f=5;
const int g=8;
```

Listato 2

```
void display7(int n)  // per pilotare i segmanti da 0-9
{
  switch(n)
  {

  case 3:
    digitalWrite(a, HIGH);
    digitalWrite(b, HIGH);
    digitalWrite(c, HIGH);
    digitalWrite(d, HIGH);
    digitalWrite(e, LOW);
    digitalWrite(f, LOW);
    digitalWrite(g, HIGH);
    break;
```

Listato 3

```
oredec=(now.hour()*100)/1000;
oreunit=((now.hour()*100)-(oredec*1000))/100;
mindec=(now.minute()*100)/1000;
minunit=((now.minute()*100)-(mindec*1000))/100;
```

Listato 4

```
display7(oreunit);
digitalWrite(disp1, HIGH);
delay(3);
digitalWrite(disp1, LOW);
display7(oredec);
digitalWrite(disp2, HIGH);
delay(3);
digitalWrite(disp2, LOW);

display7(minunit);
digitalWrite(disp3, HIGH);
delay(3);
digitalWrite(disp3, LOW);
display7(mindec);
digitalWrite(disp4, HIGH);
delay(3);
digitalWrite(disp4, LOW);
```

Listato 5

```
void setsveglia(){

  if (analogRead(A1)==LOW){ // pulsante per incrementare le ore della sveglia
    delay(200);  // pausa antirimbalzo pulsante
    oresv=oresv+1;
    if (oresv >23){ // chiaramente quando arriva a 23 riparte da zero
    oresv=0;
    }
    }

  if (analogRead(A2)==LOW){ // pulsante per incrementare i minuti della sveglia
    delay(200);
    minsv=minsv+1;
    if (minsv >59){ // chiaramente quando arriva a 59 riparte da zero
    minsv=0;
    }
    }

  oredecsv=(oresv*100)/1000;
  oreunitsv=((oresv *100)-(oredecsv*1000))/100;
  mindecsv=(minsv*100)/1000;
  minunitsv=((minsv*100)-(mindecsv*1000))/100;

   display7(oreunitsv);
digitalWrite(disp1, HIGH);
delay(3);
digitalWrite(disp1, LOW);
 display7(oredecsv);
digitalWrite(disp2, HIGH);
delay(3);
digitalWrite(disp2, LOW);

 display7(minunitsv);
digitalWrite(disp3, HIGH);
delay(3);
digitalWrite(disp3, LOW);
 display7(mindecsv);
digitalWrite(disp4, HIGH);
delay(3);
digitalWrite(disp4, LOW);

if (analogRead(A0)==LOW){ // se premuto pulsante per impostare la sveglia dopo aver
scelto l'ora
  delay(250);
EEPROM.write(1, oresv); // memorizzo nella eeprom le ore sveglia
delay(50); // piccola pausa per assicurare la corretta scrittura
EEPROM.write(2, minsv); // memorizzo nella eeprom i minuti sveglia
delay(50);
impsveglia=0; // variabile di comodo che mi permette di uscire dalla subroutine
}
}
```

ci sono 86400 secondi e l'errore è dello 0.033%, guarda un po' il caso 86400*0.033%=28.5 secondi. Alla ricerca di un compensatore da collegare tra un pin del quarzo e la Gnd ho rimediato un 10-120pf. Dopo la sua installazione e regolazione ecco il risultato (figura 9). Dopo giorni di prova ho aggiunto un altro compensatore del solito valore anche tra l'altro capo del quarzo e la GND. Con estrema sincerità vi comunico che l'errore dopo giorni di prove è costante, adesso si è portato a soli +4 sec al giorno. Le variabili che posso produrre questo errore possono essere molte, citandone alcune l'errore può venire dal mio strumento, la temperatura, dato che i quarzo è termo

Figura 7: Montaggio

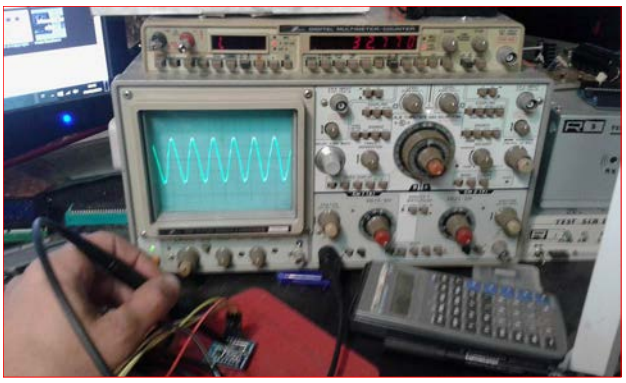


Figura 8: Primo oscillogramma

sensibile, oppure potrebbe anche essere un ripple proveniente dalla sezione alimentatore. Infastidito da questa situazione, ho lasciato perdere l'hardware. l 'errore ormai costante di 4 secondi in 24 ore l'ho risolto via software aggiungendo poche righe al sorgente (listato 6). Alle ore 19:32:04 vado a comunicare al modulo che l'ora corrente è 19:32:00. Facendo così ho corretto l'errore, però dato che l'ora torna indietro, quando l'orario ritorna alle 19:32:04, il modulo RTC subirebbe ancora un nuovo aggiornamento. L'orario rimarrebbe in questo spazio temporale di 4 secondi. Ho inserito la variabile di comodo nominata "adj". Quando la variabile ha un valore 0 avviene l'aggiornamento, altrimenti no. Come vedete dal sorgente all'inizio la variabile ha un valore 0 e avviene l'aggiornamento. Fatto l'aggiornamento cambia valore portandosi ad 1. In questo modo faccio saltare la condizione,così alle 19:32:04 l'orologio continua la sua corsa. Chiaramente la variabile và ripristinata, altrimenti dopo 24 ore siamo al solito problema. Lo faccio fare immediatamente, quando l'orario raggiunge le 19:33:00 la variabile torna a 0 e la condizione si verificherà solo quando torneranno ad essere le 19:32:04 del giorno dopo. Soluzione che

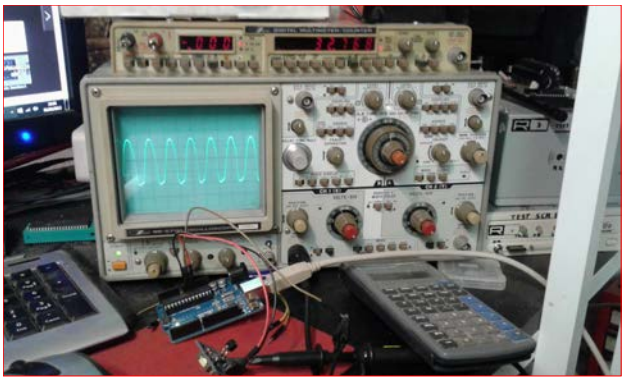


Figura 9: Secondo oscillogramma

richiede un po' di tempo e prove, oltretutto rimane come unica possibilità a chi non possiede una strumentazione affidabile. La ritengo la soluzione più efficace,ma a patto che l'errore sia costante nel tempo.

L'aggiunta finale

Durante questi giorni di prova, noto l'unico ingresso analogico libero (an3). Ho deciso di saldare con collegamenti in aria la classica sonda di temperatura LM35. Modifica al software che ad ogni minuto nell'intervallo tra i 30 e i 40 secondi il display segnala la temperatura interna per poi tornare a visualizzare l'orario corrente. Buon divertimento e buona realizzazione a tutti.

Listato 6

```
  // aggiustamento errore rtc
  while ((oredec==1 && oreunit==9 && mindec==3 && minunit==2 && secdec==0 &&
secunit==4)&&(adj==0)){
   rtc.adjust(DateTime((now.year(), DEC), (now.month(), DEC), (now.day(), DEC), 19,
32,0));
  adj=1;
      }
     if  (oredec==1 && oreunit==9 && mindec==3 && minunit==3 && secdec==0 && secunit==0){
       adj=0;
       }
```

LA NOSTRA TECNOLOGIA AL SERVIZIO DEI VOSTRI PROGETTI

Dal 1982 sviluppiamo e realizziamo i vostri progetti, attraverso la scelta dei componenti più adatti

Proponiamo una gamma completa di componenti elettronici utilizzati nei vari settori dell'industria, come telecomunicazioni, regolazione motori, gruppi di continuità e inverter, convertitori e carica batterie, automazione macchine utensili, automazione industriale, raddrizzamento, galvanica, saldatura, trazione elettrica e automotive, quadristica, fotovoltaico, eolico e altre energie rinnovabili, medicale, monitoraggio energetico, monitoraggio batterie, illuminazione a LED, segnalamento ostacolo al volo e molti altri.

CITEL · Diotec Semiconductor · LEM · SEMIKRON innovation + service · SIRIO Inductive Components · TECNOAL · QEED · FEEM · obsta · NEXTYS · RAME SERVICE

www.totemelectro.com

TOTEM ELECTRO

Viale Lombardia, 66 - 20131 Milano
tel.: 02.28.85.111 - fax: 02.28.85.11.50
totem@totemelectro.com

ARDUINO “OPTO”, l’iperbole dell’autocostruzione

di Roberto Vallini

In questo articolo tratterò approfonditamente il tema dell’autocostruzione, rapportata alle nuove tecnologie embended, cioè shield, con microprocessori incorporati e relativi moduli già pronti e configurati, pronti per l’incastro negli opportuni pin. La stazione saldante andrà in disuso? Addio “Millefori”? Forse sì, ma per il momento, c’è chi, come un reduce di guerra, porta alta la bandiera del suo reggimento e, come me, difende “l’antico profumo” di stagno fuso, di un condensatore dissaldato a fatica da una vecchia TV, per creare un lampeggiatore su una tavoletta di legno.

L’Algoritmo dell’autocostruzione

Autocostruire, per le nuove generazioni, abituate agli “starter kit”, e ai debugging dal Web, è vista, a volte, come una “cosa da romantici” superata. Essendo a contatto con i ragazzi delle scuole medie, con cui tento sempre più difficilmente di portare avanti dei progetti tecnico-ludici, come ad esempio la realizzazione di un moto inverso, o di un lampeggiatore, noto che tali temi destano sempre meno interesse. Quante volte ho sentito la frase: “Perché non creiamo un robottino? Io ho sentito che con il jostick della Play Station si può fare”. Un robottino? Un mix di meccanica teorica ed applicata, elettrotecnica, elettronica discreta, digitale, microchip, trasmissioni radio, ed una conoscenza quasi “ingegneristica” dei movimenti ortogonali, rotatori e ancora di più? Non ci siamo! Se poi prendiamo un libro di testo per gli istituti tecnici, su 440 pagine, la legge di Ohm ne occupa 4, contate. “Creare” non significa E-Bay ed una ricaricabile piena. Creare un oggetto, il quale si muova nello spazio, richiede fantasia, curiosità, ma soprattutto costanza. Occorre fare e disfare senza scoraggiarsi, per arrivare a un oggetto di cui andare fieri. Perché il problema sta proprio qui. Se, è vero che creare “modularmente” è semplicissimo, è vera la condizione per cui la cosa

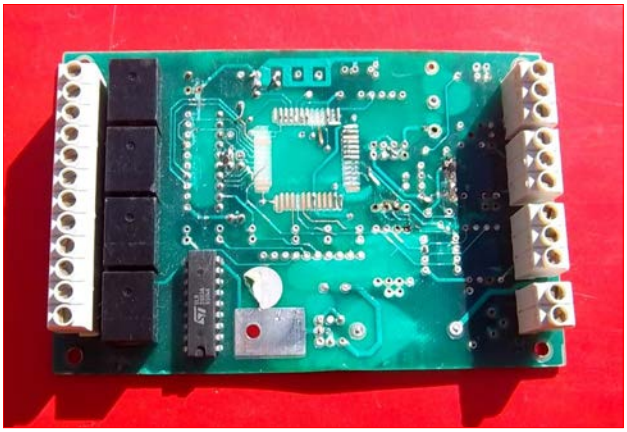


Figura 1: Tipica scheda relè dissaldata dai componenti inutili

creata annoierà presto e finirà in qualche angolo buio. Poi si passerà a fare i veri “makers”, dopo aver passato un po' di tempo sui libri, a studiare l'algebra di Boole e i principi fondamentali dell'elettrotecnica. E’ di fatti un algoritmo, più fantasia e pazienza corrispondono a una maggiore soddisfazione. Cercherò quindi, di guidare i più giovani alla “creatività totale”, attraverso lo sviluppo guidato di una semplicissima macchina, governata dai raggi infrarossi, per far scoprire, la tangente creativa che inevitabilmente interseca la “chiusa circonferenza” del “già pronto”. Inserirò poi Arduino Uno, una shield non vecchia ma nemmeno troppo nuova, cosi come il “periodo di mezzo”, che stiamo attraversando.

Robot, un po’ di storia

Il termine “robot” compare per la prima volta nel 1920, dallo scrittore ceco

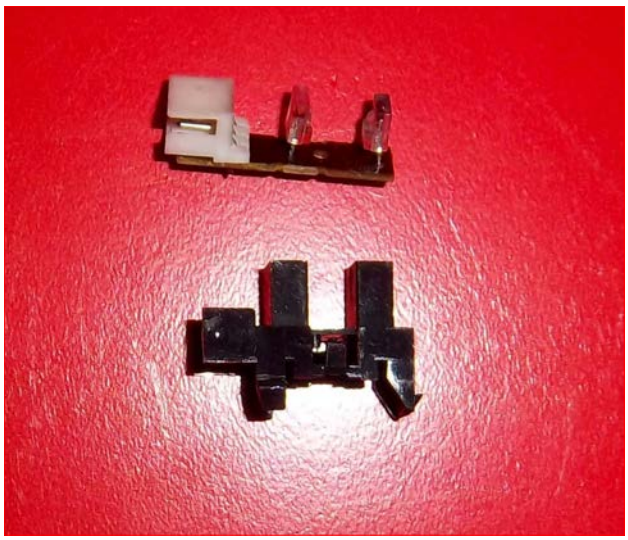


Figura 2: Fotografia dei sensori a forcella, integri e smontati

Karel Capek nel suo dramma satirico utopico R.U.R, per rappresentare il rapporto antitetico tra l'uomo e il mondo delle macchine. L'inizio del 900 vede la nascita delle prime catene di montaggio, e con esse i pareri favorevoli e contrari. Cult movie entrato nella storia con "Tempi moderni" del 1936 dove, un improvvisato Charlie Chaplin operaio, si fa trascinare in un ritmo frenetico di produzione, fino ad essere inghiottito dalla catena di montaggio. Si arrivò negli anni ottanta quando l'economista americano Jeffrey Sachs teorizzò la "fabbrica a luci spente", cioè un sito produttivo dove la presenza dell'uomo e, di conseguenza della luce, è inutile. Forse, la citazione di Furio Colombo, nello incipit del suo saggio "Confucio nel computer", definisce bene la situazione: "Come saranno la vita, le scelte, il destino del mondo digitale dei bit, che non è fisico però è reale, lungo percorsi labirintici nella rete? Ci saranno eventi nuovi, non ancora esplorati. Il nuovo, non sempre è il meglio, però è inevitabile".

Tipi di macchine

Le macchine si possono classificare in due grandi gruppi: operatrici e motrici, Con le prime, si modifica la forma o la posizione di un oggetto. Le seconde trasformano un certo tipo di energia, (chimica, elettrica, di pressione etc.) in energia meccanica utile. Sono macchine operatrici le macchine semplici, anche le leve e piani inclinati. Lo studio dei principi fisici su cui si basano è fondamentale per comprendere il funzionamento delle macchine di uso industriale. Le macchine di trasporto di persone, di merci o di lavorati, all'interno dei reparti, appartengono al gruppo delle operatrici, esse si muovono in percorsi il cui termine tecnico è "assi". Proveremo in questa descrizione a costruire una piccola macchina, la quale si muove lungo l'asse x del piano.

Modulini di ricezione e attivazione					Condizioni di saturazione BJT (con VceSat Trascurabile) valori Ib e Ic misurati (Transistor per switch) tipo 2n17111			
			Vcc (V)	I di uscita (Ib) AR1	Ic BJT (mA)	Hfe min BJT	Ic/hfe min	Ib Saturaz (mA)
IR1	Modulo di ricezione infrarossi 1	Fototransistor tipo H22A1	12	2,98mA			1,56	>1,56
AR1	Modulo di gestione E1-IR1	Relay 12V 2x3 contatti 5A, con BJT alto hfe	12		96	150		
E1	Modulo di emissione infrarossi 2	Fotodido tipo ST6019	12					
IR2	Modulo di ricezione Infrarossi 2	Fototransistor tipo H22A1	12	3,55 mA				
AR2	Modulo di gestione E2-IR2	Relay 12V 2x3 contatti 5A con BJT alto hfe	12		120	150	1,25	>1,25
E2	Modulo di emissione infrarossi	Fotodido tipo ST6019	12					

Figura 3: Tabella delle grandezze rilevate

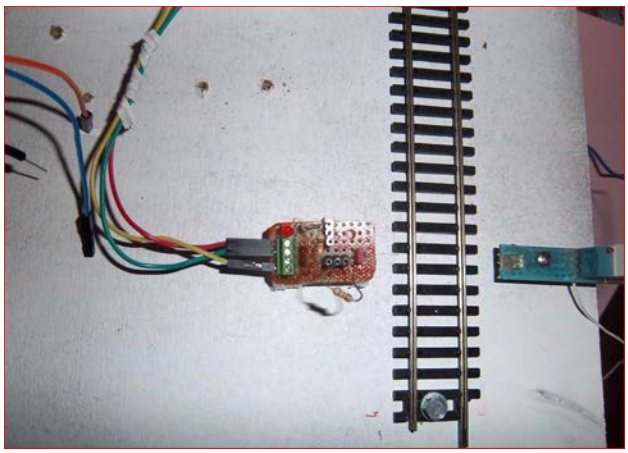


Figura 4: Moduli di ricezione-trasmissione

Un giorno, facendogli visita, vidi le teiere deturpate del coperchio, nella pattumiera. Preso da stupore, girai lo sguardo e vidi una sfilza di piccoli emisferi di ottone sul suo banco. Stava "selezionando" il giusto tetto per il suo faro in scala, da piazzare su di uno splendido diorama marino. Torniamo a noi. Costruire una macchina seppur movente su un asse richiede la lista della spesa, se siamo dei veri "auto-makers". Ecco come va impostata.

Costruire una macchina

Anche se esistono simulazioni, il tema è l'autocostruzione, quindi gli standard di progettazione vanno messi per un attimo da parte. Se, infatti, il listato risultante dal nostro layout non fosse corrispondente ai materiali e ai componenti che abbiamo in laboratorio, saremmo nei guai. Il vero auto-costruttore colleziona tutto ciò che può essergli utile in futuro. Breve inciso, ne conobbi uno, veramente bravo che, collezionava piccole teiere.

Quali sono i componenti realmente importanti?

A parte i materiali di costruzione, la parte elettronica va realizzata senza compromessi di sorta, con questo intendo dire che, non si possono inserire componenti più o meno adattabili allo scopo, a parte in alcune varianti. Se, per fare un esempio, la basetta di plexiglas su cui fissare un circuito, anche se sborda di qualche millimetro, non succede niente di grave, ne patirà il lato estetico, ma non

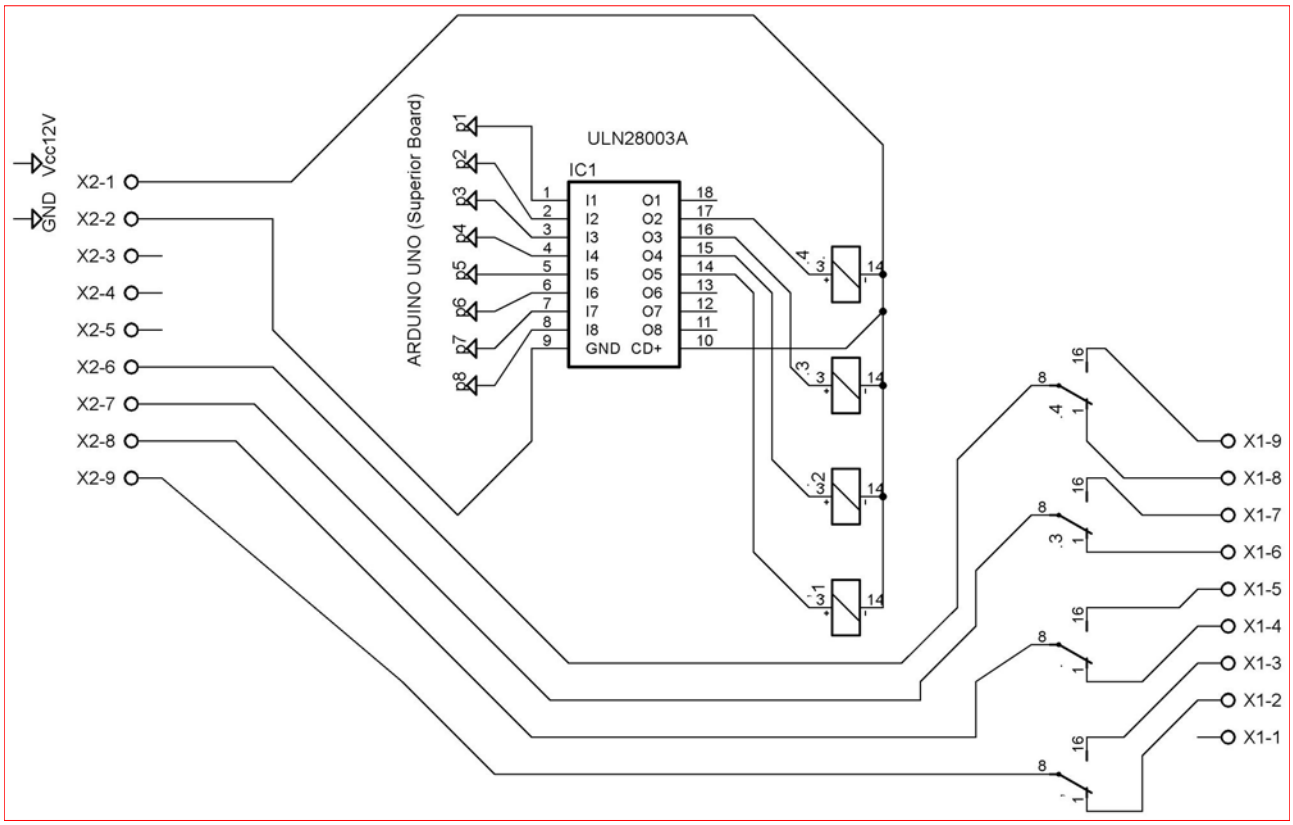


Figura 4a: Schema base di relé di scambio

impegno, etica e passione

ELSTORE

electric and electronic components

NON ACCONTENTARTI, PRETENDI IL MEGLIO

QUECTEL® Build a Smarter World — MODULI GNSS GSM/UMTS/LTE

POWERTIP — DISPLAY LCD

ASAHI KEIKI — TERMOSTATI BIMETALLICI

CDTECH 思迪科 — DISPLAY LCD

SAURO ELECTRONIC CONNECTORS by the law — CONNETTORI E MORSETTI

avalue Technology Inc. — SISTEMI EMBEDDED

ZIPPY — MICROINTERRUTTORI

Well® — DISPLAY LCD

Elstore s.r.l.
Via Francesco Crispi, 40A 43126 Parma (PR)
Tel: 0521-672532 Fax: 0521-671179
Email: commerciale@elstore.it
Web: www.elstore.it

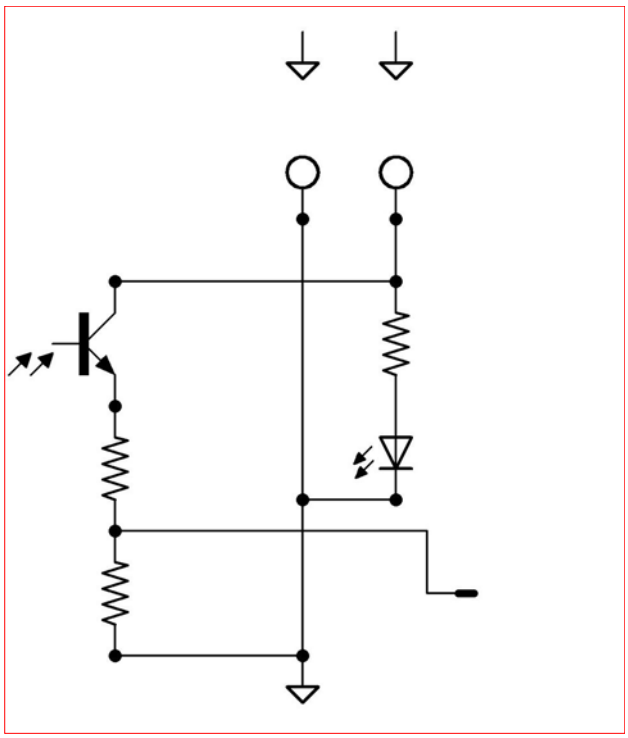


Figura 5: Schema dei moduli di ricezione-trasmissione

il funzionamento dello stesso. Per i componenti elettronici il discorso cambia, un circuito progettato ha quei componenti e altri non possono essere. Nell'ottica dell'autocostruzione i componenti possono essere recuperati da schede in disuso, dissaldandoli con molta cautela. E' un'operazione che, richiede tempo e pazienza, anche qui occorre chiedersi: "Ma ne vale la pena"? Ne vale la pena nella misura in cui, il tempo trascorso a dissaldare, supera il costo del componente stesso. Occorre individuare dei tipi di componente che potranno servire per altri progetti e quelli di maggior costo commerciale.

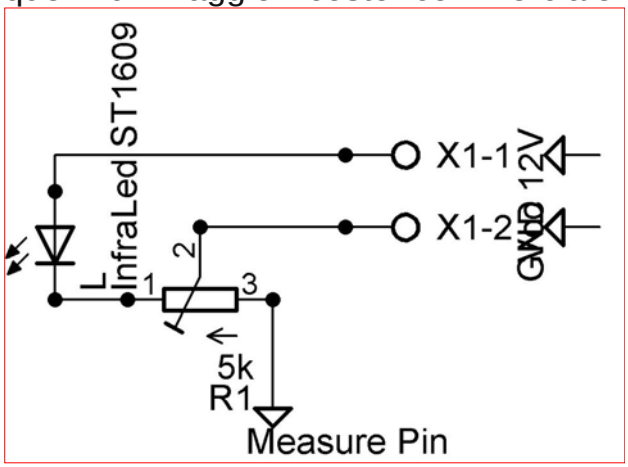


Figura 5a: Schema modulo emettitore

Innanzitutto la serie LM317. Questo integrato, ha le dimensioni di un normale transistor di potenza tipo TO220, dispone di tre piedini, ed è utilizzato come stabilizzatore di tensione, erogatore di corrente, variatore e, altri molteplici usi. A seguire, la serie 78xx, dove le due x vengono sostitute con il livello di tensione da stabilizzare. Trattasi infatti, anche in questo caso di un integrato con tre piedini, anch'esso tipo TO220, il quale livella il valore nei circuiti, multi tensione. Un'applicazione molto sfruttata del 7805, è quella dove, su di un'unica scheda relè, alimentata a 12V, occorre avere una derivazione a 5V per gli integrati TTL. Immancabile la serie dell'integrato ULN28003A, nello specifico, un array di 8 Darlinghton, Common Emitter, dal quale datasheet, si evince che, supportano una massima corrente di ben 500 mA, ideale per dispositivi di potenza media, (come premesso Arduino non supportando correnti importanti, necessita dei "muletti" capaci di implementare una corrente superiore ai 40mA). Ultimo, non per importanza, l'integrato 555, con cui si possono creare multivibratori di ogni tipo, compresi timer e generatori di onde. Per concludere, direi che i relè, nello specifico quelli con bobina da 5 o 12V, visto il costo, non vanno di certo buttati nella spazzatura. Il tempo è prezioso, anche per un auto-makers e forse è meglio dare un'occhiata al Web.

Cominciare dal semplice

Partiremo da un carro, useremo una parte di una vecchia locomotiva giocattolo, posizionato su una rotaia a due poli, che danno energia al piccolo motoriduttore, la quale, emulando il movimento sull'asse "X", compirà una routine di trasporto, durante la sua routine operativa, in un'ipotetica officina di produzione al sito di

stoccaggio. Lungo il suo percorso, la nostra "macchina" incontrerà dei sensori ottici ad infrarossi (diffusissimi nelle industrie) in quanto, garantiscono, oltre il funzionamento del processo, una sicurezza per gli operatori. Gli operai, se si trovassero accidentalmente sul percorso del transponder, bloccherebbero il ciclo, fino a riarmo. Nella mia protoboard ho usato, smontandoli (nella logica dello scoprire) dei sensori ottici a forcella, per l'affidabilità dei componenti interni, visto che tentando con foto-resistenze e fotodiodi tradizionali ho riscontrato un "inquinamento" non indifferente, dato dalla luce diurna o artificiale.

Breve descrizione dei raggi infrarossi e di sensori a forchetta

I sensori a forcella sono composti da un emettitore di raggi infrarossi posto in asse con un ricevitore sensibile a tale emissione luminosa. Sono posti e saldati su di un unico circuito, e ricoperti da una struttura plastica nera con due feritoie specchiate, in prossimità del fascio luminoso. L'emettitore è solitamente un fotodiodo, ovvero un Led I.R. Il ricevitore è un fototrasnsistor, collegato a "base aperta", ovvero il collettore riceve l'alimentazione positiva, l'emettitore è a massa come il catodo del diodo contrapposto e, la base non ha terminali. Il diodo, con l'anodo al positivo, se non riceve tensione non emette nessun tipo di luce. Una volta alimentato, il led I.R. sfrutta le capacità ottiche di alcuni metalli ionizzati, per emettere fotoni, secondo il noto principio fisico dell'emissione spontanea. Stesso fenomeno, però, fisicamente rovesciato, accade sulla base del fototransistor la quale, ricevendo i fotoni, emette una piccola corrente, che scorre tra collettore ed emettitore. Se poi tale segnale viene amplificato

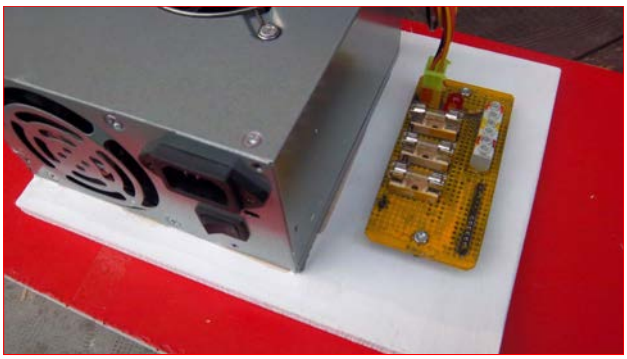


Figura 6: Questa è la fotografia dell'ATX "modificato"

opportunamente, si ottiene un dispositivo che inserito in un ambito di potenza, funziona a tutti gli effetti, come un interruttore ottico, poco sensibile alla luce visibile.

Sensori come interruttori

Nel progetto, ho predisposto sei moduli, con le caratteristiche riportate in figura 2. Ho innanzitutto creato due emettitori d'infrarosso, denominati E1 (emettitore 1) ed E2 (Emettitore 2), nello specifico due Led I.R. posti su una basetta con una resistenza variabile di limitazione di corrente di 5Kohm. Gli emettitori sono stati estratti dal loro contenitore e rimessi in asse. Questa operazione è stata necessaria perché lo scarto dei binari era di gran lunga superiore ai 5 millimetri della "forcella". E1 ed E2, una volta alimentati, creano un fascio di fotoni continuo, il quale, va a colpire il fototransistor accoppiato: (IR1-IR2) Entrambi i fototransistor, hanno ricevuto la stessa sorte dei fotodiodi,

Figura 7: Base relè finita

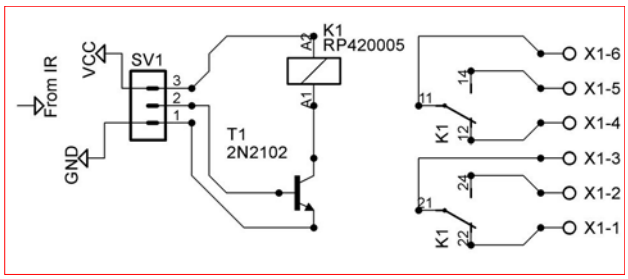


Figura 7a: Schema modulo relé

sono stati collegati a una Vcc di 12 V, con una resistenza di limitazione di 10 Kohm, ed una resistenza di polarizzazione in base di 10Kohm. Infatti, dal collettore del foto transistor, tramite opportuno collegamento, mi sono diretto alla base di un transistor, capace di amplificare il debole segnale, entrando nello stato di saturazione, o viceversa (non ricevendo alcuna corrente) in interdizione. Ho scelto un transistor 2N17111, per le sue peculiarità di utilizzo, in condizioni non lineari. Mi necessitava un transistor, che, amplificando la debole corrente in arrivo alla sua base, si ponesse in condizione di saturazione, per attivare la bobina del relè, e di conseguenza arrivare ad Arduino con dei segnali "puliti" (alto o basso). Non mi sono fidato delle mille configurazioni che si vedono qui e là sul web, con bjt che interagiscono con Arduino, tramite polarizzazioni non sempre affidabili. I relè scelti hanno 8 pin di cui 2 per la bobina e 2 contatti in scambio separati. La resistenza della bobina è di 175 ohm quindi, alimentati a 12V assorbono 0,07A dissipando 0,86W. Occorreva un transistor robusto, con

Listato 1

```
//Questa routine prevede uno scambio di polarità//
//tra i binari, condizionata dai due fotosensori lungo il percorso//

void setup()
{
   pinMode(2,OUTPUT);   //Collegamento a binario A//
   pinMode(3,OUTPUT);   //Collegamento a binario B//
   pinMode(4,OUTPUT);   //Collegamento a binario B//
   pinMode(5,OUTPUT);   //Collegamento a binario A//
   pinMode(9,INPUT);    //Segnale proveniente da AR1//
   pinMode(10,INPUT);   //Segnale proveniente da AR2//
}

void loop(){
   int ValPin9=digitalRead(9);    //leggi I valori//
   int ValPin10=digitalRead(10);  //leggi I valori//
   digitalRead(9);
   digitalRead(10);
   if((ValPin9==HIGH)&&(ValPin10==LOW))  //Se AR1 è alto//
   {                                     //A=tensione positivae B=tensione negativa//
      digitalWrite(2,HIGH);              //Quindi il carro va dal punto A, al punto B//
      digitalWrite(4,HIGH);
      digitalWrite(3,LOW);
      digitalWrite(5,LOW);
   }
   if((ValPin9==LOW)&&(ValPin10==HIGH))  //Se AR2 è alto//
   {                                     //A=tensione negativa B=tensione positiva//
                                         //quindi inversione di marcia//
      digitalWrite(2,LOW);
      digitalWrite(4,LOW);
      digitalWrite(3,HIGH);
      digitalWrite(5,HIGH);
      delay(5000);
   }
}
```

SEGUICI SU facebook

Clicca "Mi Piace"
e scarica subito la
Guida Android

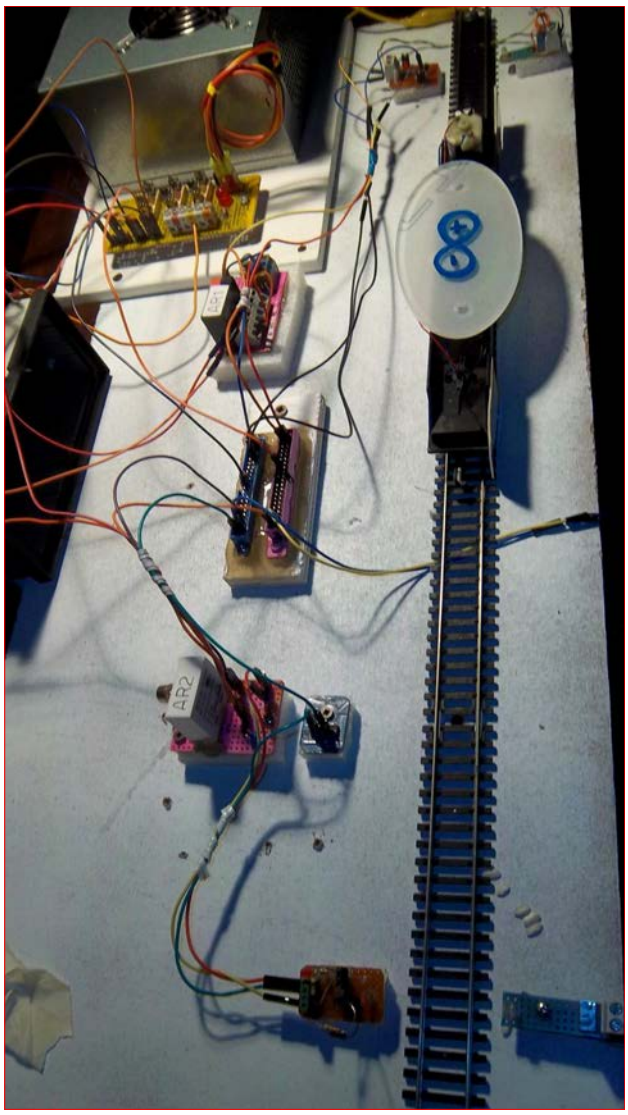


Figura 8: Questa è protoboard per esperimenti ottici e di movimento con Arduino

la mia scelta penso di esser andato sul sicuro. Viste le dimensioni del progetto si potevano utilizzare relè più piccoli ma, nella filosofia auto-costruttiva, questi due componenti, sono stati riparati e resi utili anche per progetti futuri.

Posizionamento e realizzazione

Come si è visto nelle figure 3,4 e 5 tutto è stato auto-costruito è si è fatto ampio uso di basette millefori. Per poter alimentare il progetto, è stato riciclato e messo a nuova vita un alimentatore ATX per computer fissi. Nel caso dell'ATX, che ben tutti conosciamo, è stato dotato, delle tre tensioni necessarie per il

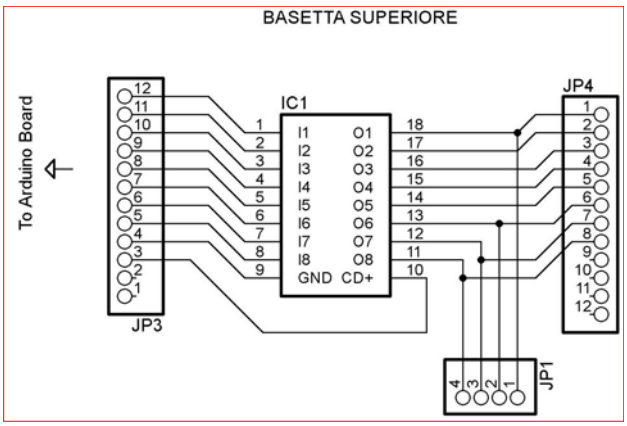


Figura 8a: Schema collegamenti Base superiore

funzionamento prototipale: 3,3 Volt per il carro, 5 Volt per Arduino Uno, e 12 Volt per la funzionalità dei relè. Se osservate la figura 5 noterete che, con una profonda opera di dissaldo, sono stati eliminati 16 conduttori sui 20 destinati al connettore per la motherboard. Il cavo color verde, destinato all'avviamento secondario, è stato cortocircuitato a massa, in maniera che l'interruttore generale, attivi tutto il sistema. Quindi, con il connettore quadripolare ho costruito una shield TRE con tipi di connessione: Morsettiera, pin header maschio e pin header femmina, logicamente collegati in parallelo e, tutti e tre "fu sibilati" separatamente. Tutto questo per poter usufruire di un alimentatore veramente sicuro, ma soprattutto "custom".

Arduino Uno entra in campo

Adesso Ora entra in campo il nostro ATmega328, il quale deve governare il sistema. Ho usato una configurazione con componenti recuperati. Ho utilizzato una vecchia scheda con 4 relè 12V-0,3W, con un contatto in scambio. Visto che di tali schede ne ho alcune, ho recuperato 4 relè, un ULN28003A, e le morsettiere di I/O. Ho realizzato un "Arduino PLC" a tutti gli effetti. Certamente sperimentale ma dotato di quattro livelli sotto indicati:

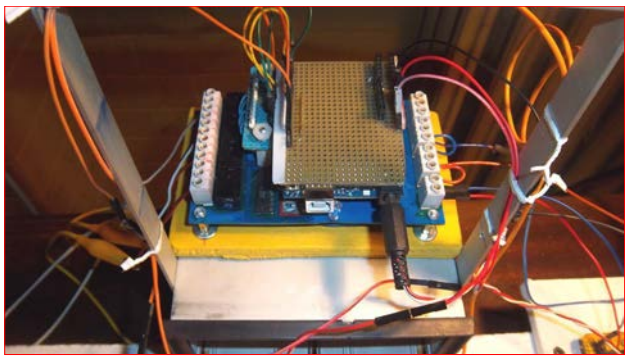


Figura 9: Foto Arduino PLC

1° Livello: base relè, con morsettiera, ULN 28003A e illuminazione blu, per dare un tocco di eleganza;

2° Livello: Posizionamento di Arduino Uno;

3° Livello: Inserimento di una mini protoboard autocostruita, con morsettiera con i pin corrispondenti agli ingressi ed alle uscite dell'ULN28003A. Questa finezza, che mi è costata un grande lavoro, permette alcune agevolazioni. La prima, di poter rispettare il pin-out di Arduino senza problemi, senza cavi svolazzanti. La seconda di pilotare in cascata altri relè visto che, per questa prima parte ne necessitavano solamente 4, ed essendo l'integrato, predisposto per 8 ingressi/uscite, ne avanzavano 4, perché non sfruttarle?

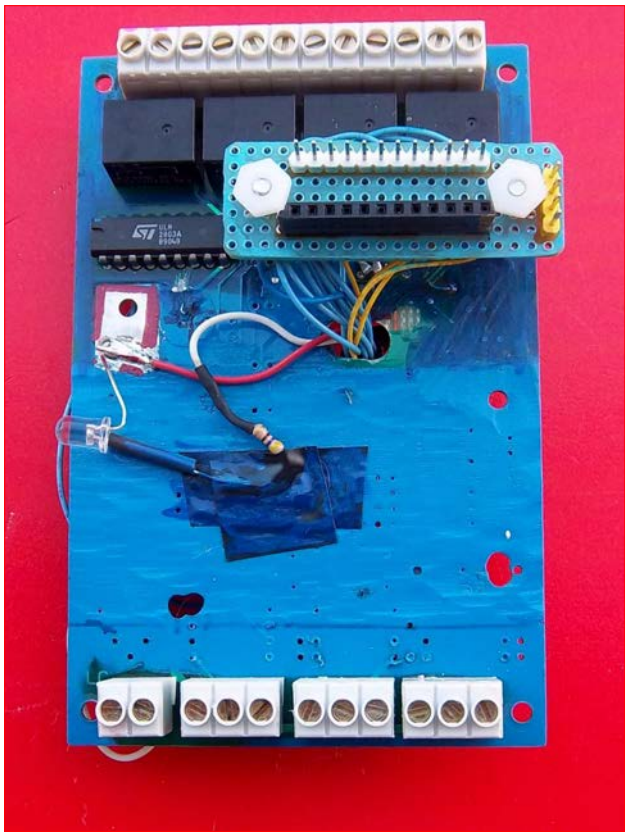


Figura 10: Scheda completa

4° Livello: Una ulteriore protoboard, per agevolare esperimenti, ricavata da una millefori a doppia faccia, ottenuta con un lungo lavoro, per via dello sfalsamento della pin-header di Arduino Uno.

Su questa questione, si è dibattuto molto sui vari forum, ma finora nessuno ha saputo dare una risposta a questa complicanza, la quale obbliga, se si vuole autocostruire una proto-Board, ad una fresatura di precisione con relativo incollaggio dei pin sfalsati. Quest'ultima, illustrata in figura, è stata la mia soluzione.

Conclusioni

Questo progetto, come premesso nell'incipit, è stata una sfida tra due mondi, quello delle release, frutto di schemi e progetti già pronti da incastrare, (embedded) ed il mondo della "creazione totale". Da tutte le immagini, capirete come, sono state impegnate ore e ore, per riuscire a far combaciare oggetti, nati ognuno per un'applicazione tipica, trasformarli e renderli utili, per qualcosa di mio, e solo mio, che potrà avere sviluppi futuri, con aggiunte di nuovi "device" e nuovi movimenti. Ora la mia "Protoboard per robot" è stata creata, questa è la genesi dei movimenti, delle leve, delle rotazioni. Lo dedico ai più giovani, con lo scopo che possano comprendere che "il robottino", sempre come citavo all'inizio, te lo devi sudare, avendo l'umiltà di capire quante variabili occorre considerare per un moto inverso, Poi si passerà a fare i veri "makers", dopo aver trascorso un po' di tempo sui libri, a studiare l'algebra di Boole, e i principi fondamentali dell'elettrotecnica.

DIRETTORE RESPONSABILE
Antonio Cirella

DIRETTORE TECNICO
Giovanni Di Maria

Hanno collaborato in questo numero:
Ivan Scordato, Grazia Ancona
Alberto Trasimeni, Giovanni Carrera
Girolamo D'Orio, Roberto Vallini.

Direzione Redazione:
CONSORZIO TECNOIMPRESE SCARL
Via Console Flaminio, 19
20134 MILANO – MI

fare elettronica

IE cloud

© Copyright
Tutti i diritti di riproduzione o di traduzione degli articoli pubblicati sono riservati.
Manoscritti, disegni e fotografie sono di proprietà di **CONSORZIO TECNOIMPRESE SCARL**. È vietata la riproduzione anche parziale degli articoli salvo espressa autorizzazione scritta dell'editore. I contenuti pubblicitari sono riportati senza responsabilità, a puro titolo informativo.